infrastructures

MDPI

Article

Genetic Algorithm for Inspection and Maintenance Planning of Deteriorating Structural Systems: Application to Pressure Vessels

Shane Haladuick and Markus R. Dann *

Department of Civil Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; smhaladuick@gmail.com

* Correspondence: mrdann@ucalgary.ca; Tel.: +1-403-220-5328

Received: 18 June 2018; Accepted: 27 August 2018; Published: 30 August 2018

Abstract: For engineering systems, decision analysis can be used to determine the optimal decision from a set of options via utility maximization. Applied to inspection and maintenance planning, decision analysis can determine the best inspection and maintenance plan to follow. Decision analysis is relatively straightforward for simple systems. However, for more complex systems with many components or defects, the set of all possible inspection and maintenance plans can be very large. This paper presents the use of a genetic algorithm to perform inspection and maintenance plan optimization for complex systems. The performance of the genetic algorithm is compared to optimization by exhaustive search. A numerical example of life cycle maintenance planning for a corroding pressure vessel is used to illustrate the method. Genetic algorithms are found to be an effective approach to reduce the computational demand of solving complex inspection and maintenance optimizations.

Keywords: risk-based inspection and maintenance; lifecycle analysis; genetic algorithm; large-scale optimization; pressure vessel

1. Introduction

In lifecycle engineering [1], the optimal engineering solution is the one that maximizes the utility provided by the system, where utility is a measure of the preference of the relevant stakeholders. Applied to inspection and maintenance planning, lifecycle engineering can be used to determine the optimal inspection and maintenance plan for a structural system. Due to the uncertainties involved in predicting structural system failure, a probabilistic risk-based approach is typically taken, where the expected value of the utility is optimized. This approach is termed "decision analysis", and was first introduced by Von Neuman and Morgenstern [2], and expanded upon in many texts since [3–6]. The application of decision analysis to the fields of inspection planning and maintenance planning are called "risk-based inspection" (RBI) [7] and "risk-based maintenance" (RBM) planning [8], respectively.

The goal of RBI is typically to optimize the timing and type of inspections, and likewise the maintenance actions for RBM. RBI and RBM have been extensively applied to deteriorating structures, such as offshore structures [9], steel structures [7,10], floating production, storage, and offloading facilities [11,12], pipelines [13–17], refinery piping [18], bridges [19,20], nuclear power plants [21], and processing plants [22].

For structural systems with independent failure, inspection, and maintenance events, performing RBI or RBM is relatively straightforward, as each component or defect can be assessed individually to determine the optimal plan. However, most structural systems (e.g., pressure vessels, bridges, and power plants) do not have independent failure, inspection, and maintenance events [23].

The failure of one component is typically related to the failure of other components and the whole system (e.g., a bridge support failing leading to bridge failure). Inspection and maintenance costs for one component are typically not independent of other inspections or maintenance, as there is often a cost savings in inspecting or maintaining multiple components simultaneously. For systems with dependent failure, inspection, or maintenance events, the optimal time and type of inspection and maintenance cannot be determined independently for each component or defect in the system. Instead, one optimal plan must be determined for the entire system [24]. The set of all candidate inspection and maintenance plans is known as the solution space, and for systems with many inspection and maintenance times or types and components or defects, the solution space can be very large. The most obvious approach to determine the optimal plan from a solution space is an exhaustive search of the entire solution space to find the plan with the greatest utility. However, for sufficiently complex systems, the solution space of all combinations of inspection and maintenance times and types becomes very large, and the exhaustive approach becomes computationally expensive and inefficient.

As a workaround, many RBI and RBM studies propose simplifications to restrict the size of the solution space, such as the following:

- Restricting the set of inspections to allow only one or two inspections over the lifetime of a system [13,17];
- Restricting inspections to a fixed time interval [7,15,16,20];
- Using a constraint (e.g., a reliability or risk constraint) to reduce the set of inspection times [7];
- Restricting the number of components or defects in the system [7,11,14–16,25];
- Treating the inspection, maintenance, and failure events for each component or defect as independent, so the optimal inspection and maintenance plan can be determined for each defect individually [12,13,26,27];

Restricting the solution space reduces the computation time for an analysis. However, it also reduces the quality of the final solution since the true optimal solution may not be in the restricted solution space. An alternative to restricting the solution space is to use a heuristic algorithm. Heuristic algorithms such as genetic algorithms [28,29] or simulated annealing [30] are well-established in computational engineering, but their adoption in RBI and RBM has been rare. Barone and Frangopol [20] used a genetic algorithm to determine the optimal inspection frequency for a deteriorating bridge. Fujimoto et al. [10] used a genetic algorithm to determine the optimal inspection times and inspection type of fatiguing structures with single or multiple components. They simplified the framework by ignoring the failure cost and instead using a reliability constraint. Martorell et al. [21] used a genetic algorithm to optimize the inspection interval in a nuclear power plant. Marseguerra and Zio [22] used a genetic algorithm to optimize both the inspection interval and the type of maintenance for several series-parallel systems, including a chemical processing plant.

The objective of this study is to develop a framework for, and examine the performance of, a genetic algorithm used to solve an optimization problem in RBI or RBM. Both the computational efficiency and the accuracy of the final solution are assessed to examine the value of using a genetic algorithm for the RBI or RBM of complex structural systems. This study purposefully keeps the methodology as generic as possible so that it can be applied to many different systems. Evaluations of appropriate ranges for the model parameters are performed to facilitate easy application of this method in the future.

This paper comprises five sections. The second section formulates the generic RBI and RBM objective function to be optimized. A corroding pressure vessel is used as an example to demonstrate the derivation of a specific objective function. The third section introduces the genetic algorithm used to solve the optimization problem. The fourth section presents the optimization for the corroding pressure vessel system and examines the results for examples with both a small and large solution space. Finally, the impact and limitations of the methodology are discussed.

2. Developing the Objective Function

2.1. Generic Objective Function

The objective function of the expected utility $\mathrm{E}[U]$ of a system is equal to the expected benefit $\mathrm{E}[B]$ provided by the system minus all of the expected costs of the system over its lifecycle [1]. The cost typically includes the cost of initial construction C_C, inspection C_I, maintenance C_M, failure C_F, and environmental pollution C_P. The failure cost includes both the direct and indirect costs of failure. The direct costs include the cost of repairing or replacing the system, cleaning up the failure, losses of having the system offline, environmental costs, and societal costs of potential human casualties. The indirect costs include any far-reaching costs of the failure. For example, for a bridge it could be failure of the greater transportation network and also damage to the reputation of the companies, government, or industry involved. Some of these costs are non-monetary, such as human casualties and environmental damage. However, it is assumed that these attributes can be monetized for comparison with other costs [31]. The expected utility $\mathrm{E}[U(\mathbf{e})]$ is given by the following equation:

$$\mathrm{E}[U(\mathbf{e})] = \mathrm{E}[B(\mathbf{e})] - \mathrm{E}[C_C(\mathbf{e})] - \mathrm{E}[C_I(\mathbf{e})] - \mathrm{E}[C_M(\mathbf{e})] - \mathrm{E}[C_F(\mathbf{e})] - \mathrm{E}[C_P(\mathbf{e})], \tag{1}$$

where $\mathbf{e}$ is the vector of all inspection and maintenance plan parameters to be optimized. The objective of the optimization problem is to determine the inspection and maintenance plan $\mathbf{e}$ that maximizes the objective function over the system lifecycle.

For a generic structural system, each element in Equation (1) must be treated as dependent on the inspection and maintenance plan $\mathbf{e}$. For instance, more frequent or higher-quality inspections or maintenance increases the expected inspection cost $\mathrm{E}[C_I(\mathbf{e})]$ and the expected maintenance cost $\mathrm{E}[C_M(\mathbf{e})]$, and decreases the expected failure cost $\mathrm{E}[C_F(\mathbf{e})]$. However, it can also potentially affect the expected pollution cost $\mathrm{E}[C_P(\mathbf{e})]$, because better-maintained components may pollute less or more. The construction cost $\mathrm{E}[C_C(\mathbf{e})]$ may also be dependent on the inspection and maintenance plan—for example, more advanced construction techniques may be required for more advanced inspections and maintenance. Finally, the expected benefit $\mathrm{E}[B(\mathbf{e})]$ is dependent because some systems produce a greater benefit when they are better maintained. Additionally, when maintenance or failure occurs, the system will likely be taken offline so that it can be renewed, resulting in a decrease in the benefit. For a generic system, these dependencies cannot be relaxed. However, for many systems they are not necessary, so they need to be examined on a case-by-case basis.

2.2. Objective Function for the RBM of a Corroding Pressure Vessel

This paper applies RBM to optimize the repair time for a set of corrosion defects in a pressure vessel. Since the objective of the RBM is to optimize the repair time for each defect, the maintenance plan $\mathbf{e}$ is defined as a vector of repair times $\mathbf{e} = (\tau_1, ..., \tau_J)^{\mathrm{T}}$, where $J > 1$ is the number of defects in the pressure vessel. The objective function for a specific system is derived by starting with the generic objective function (1) and relaxing the dependencies on the inspection and maintenance plan as applicable.

For pressure vessel RBM, several costs in (1) can be assumed to be independent of the maintenance plan $\mathbf{e}$. First, the benefit of a pressure vessel is independent of its state of deterioration, given that failure has not occurred. There is still the lost benefit due to system's offline time during maintenance and failure. However, this loss can be incorporated into the cost of maintenance and failure. Consequently, the benefit is independent of the inspection and maintenance plan $\mathbf{e}$. Second, the construction cost of a pressure vessel is typically independent of the inspection and maintenance plan. Third, pressure vessels do not typically pollute, again given that failure has not occurred, so the pollution is independent of the state of deterioration. Finally, the cost of inspection is independent of the maintenance plan, because RBM is performed to optimize repair times and the vessel will still be inspected according to a separate inspection plan. Costs that are independent of the maintenance plan are constant for all possible maintenance plans, and therefore those costs can be ignored when comparing the different plans. The objective function in (1) can be re-written for the pressure vessel systems in terms of the expected cost $\mathrm{E}[C(\mathbf{e})]$:

$$\mathrm{E}[C(\mathbf{e})] = \mathrm{E}[C_M(\mathbf{e})] + \mathrm{E}[C_F(\mathbf{e})] + \text{constant}, \tag{2}$$

where the constant summarizes all of the costs that are independent of $\mathbf{e}$. The objective now becomes finding the optimal maintenance plan $\mathbf{e}$ that minimizes the expected cost $\mathrm{E}[C(\mathbf{e})]$ over the lifecycle of the pressure vessel.

The expected costs of maintenance and failure are required in (2) to perform the RBM. As discussed previously, system-wide RBM optimization is only warranted when there is dependency in either the maintenance or failure events between the defects. The dependency of the maintenance and failure events must reflect the physical nature of the system. In this example, the maintenance events are treated as dependent and the failure events as independent based on the following justification. When repairing a pressure vessel, it is often the case that the bulk of the cost is due to the pressure vessel being offline, emptying it, and opening it up for repair. This cost is a one-time cost, regardless of the number of defects being repaired. There is an additional cost of the actual repair for each defect. The dependent maintenance events allow the repair cost to account for this cost relationship by defining the repair cost as a function of the number of simultaneous repairs. Any function can define the repair cost. For example, a non-linear function of the number of repairs allows the cost of the first repair to be greater than subsequent repairs. The expected cost of maintenance is given by the following:

$$\mathrm{E}[C_M(\mathbf{e})] = \int_0^{t_{sl}} \sum_{k=1}^{K(t)} \frac{C_R(k)}{(1+r)^t} p_R(\mathbf{e}, t, k)\mathrm{d}t, \tag{3}$$

where t_{sl} is the service life of the pressure vessel, $C_R(k)$ is the cost of repairing k defects, $K(t)$ is the number of simultaneous repairs scheduled at time t, r is the discount rate, and $p_R(\mathbf{e}, t, k)$ is the probability of repairing k defects at time t for maintenance plan $\mathbf{e}$. The summation (3) is performed over the number of possible repairs at time t, which is from 1 to the number of repairs $K(t)$ that are scheduled at time t. All $K(t)$ repairs that are scheduled at time t will not necessarily be undertaken, because there is the potential for any number of the $K(t)$ defects to fail before the repair. The integration in (3) is performed from now ($t = 0$) to the service life t_{sl} to consider any continuous potential repair time. The repair cost is discounted by r, yielding the present value of the cost. The probability $p_R(\mathbf{e}, t, k)$ of repairing k defects at time t for maintenance plan $\mathbf{e}$ is given by the following:

$$p_R(\mathbf{e}, t, k) = \sum_{S \subseteq L} \left[\prod_{j \in S} (1 - F_j(\mathbf{e}, t)) \prod_{j \in L, j \notin S} F_j(\mathbf{e}, t) \right]. \tag{4}$$

The summation in (4) is performed over all possible combinations of defect subsets S of L, where each S is a subset of k defects selected from the set L of $K(t)$ defects that are scheduled for repair at time t. The first product in (4) is performed over the subset S of defects that are repaired. The term $1 - F_j(\mathbf{e}, t)$ is the probability that defect j survives until the repair at time t, and $F_j(\mathbf{e}, t)$ is the cumulative distribution function (CDF) of the time to failure T_j of defect j. The second product is performed over the defects that are in the set L that are scheduled to be repaired at t, but not in the subset S that are actually repaired, and the term $F_j(\mathbf{e}, t)$ is the probability that defect j fails before it is repaired at time t.

The failure events are assumed to be independent to reflect a pressure vessel susceptible to leak failure with no risk of burst failure. A leak at one defect does not affect the other defects, and thus the failures can be treated as independent. The expected cost of failure is determined using the following equation:

$$\mathrm{E}[C_F(\mathbf{e})] = \int_0^{t_{sl}} \frac{\mathrm{E}[n_F(\mathbf{e}, t)]C_F}{(1+r)^t} \mathrm{d}t, \tag{5}$$

where $\mathrm{E}[n_F(\mathbf{e}, t)]$ is the time-dependent expected number of failures in for the repair plan $\mathbf{e}$, and C_F is the cost of a single leak failure. It is given by the sum of the CDF of time to failure for each defect:

$$\mathrm{E}[n_F(\mathbf{e}, t)] = \sum_{j=1}^{J} F_j(\mathbf{e}, t), \tag{6}$$

where $F_j(\mathbf{e}, t)$ is the CDF of time to failure for defect j based on maintenance plan $\mathbf{e}$. The objective function (2) is evaluated for each possible repair plan by populating Equations (3) and (5) using the expected number of failures in (6).

3. Optimization with a Genetic Algorithm

In general, repair times can be selected from any continuous time throughout the lifecycle of a system. However, allowing continuous repair times creates a continuous optimization problem with an infinitely large solution space of candidate maintenance plans. As a simplification, repair times are restricted to a discrete set of potential repair times throughout a system's lifecycle. A short time interval between the potential repair times minimizes the impact of the simplification, and is in line with common practice as repair times are not optimized to the nearest day. Letting $m > 1$ denote the number of potential repair times over a lifecycle, plus the option to not repair, the solution space contains $(m+1)^J$ candidate repair plans, which are the combinations of repairing each defect at any potential time. Determining the optimal inspection plan $\mathbf{e}$ from the set of $(m+1)^J$ candidate inspection plans is a discontinuous or integer optimization problem [32]. Many optimization techniques are not applicable to integer optimization problems. However, genetic algorithms have been shown to be successful. This section describes the use of a genetic algorithm to perform the RBM optimization for a corroding pressure vessel.

A genetic algorithm is a heuristic based on biological evolution [33]. Evolution relies on the processes of natural selection and mutation to evolve a child population that is better adapted to its environment than the parent generation. Evolution begins with an initial population possessing variation in their genetic traits. Some members of the population are better adapted than others, and these members are more likely to survive and reproduce. When members reproduce, there is heredity in the reproduction process, meaning that the genetic traits of the parents are passed to their children. The genetic traits of a child are composed of a random crossover of the parents' traits. Preferential reproduction means that the genetic traits of the fitter members are preferentially passed on to the next generation. The traits of a child can also randomly mutate, allowing the child to possess traits that were not present in either parent. Through an iteration of the processes of natural selection and mutation, a fitter population evolves over many generations. Genetic algorithms replicate the evolutionary process. This study uses the genetic algorithm method described in [34] for solving integer optimization problems. In the pressure vessel example, the members of the population are the candidate solutions, and the genetic traits of each member are the repair times for each defect. The generic algorithm iterates successive generations of the same population size, where the best member of each generation progresses towards the optimal solution, based on the applied objective function. Starting with a randomly selected population of candidate solutions, the objective function is evaluated to rank the fitness of each solution. The best candidates from the population are termed the elites and are passed onto the child generation unevolved. To produce the remaining members of the child population, the genetic algorithm replicates the processes of crossover and mutation. Crossover children are created by combining the genes of two parent members. A crossover function specifies which traits are inherited from each parent. In the pressure vessel problem, a crossover child has repair times for some of the defects based on one parent, and the repair times for the remaining defects based on the other parent. Crossover instills heredity, allowing the genes of the fitter members of the parent generation to be passed onto the child generation. Mutation children are created by randomly mutating the genes of a single parent member. A mutation function defines which parent traits are mutated as well as the degree of mutation.

In the pressure vessel problem, a mutation is a random change of the repair time for a defect. Mutation allows the genetic algorithm to search candidate solutions that are not part of the initial population, promoting a wider search for the globally optimal solution. However, excessive mutation makes the algorithm less efficient, because it randomly changes solutions that are already well adapted. A crossover rate is used to determine the proportion of the remaining children that are produced by crossover and mutation. At one extreme, when all children are produced from crossover, the algorithm risks becoming trapped in a locally optimal solution, because it cannot mutate new genes that were not in the original population. At the other extreme, when all children are mutated, the algorithm performs an exhaustive search, because there is no heredity. A selection function determines which of the members of the parent population are used for crossover and mutation based on the objective function score for each parent member. By iteratively repeating this process for many generations, the genetic algorithm evolves the population towards fitter solutions, and the best solution from each generation progresses towards the optimal solution. A criterion is used to terminate the algorithm, which is typically either a maximum number of generations or computation time, or a maximum number of generations without improvement in the score of the best solution. Obtaining the optimal solution is not guaranteed.

4. Numerical Example of a Corroding Pressure Vessel

To demonstrate the methodology, two numerical examples are presented: one with a small and one with a large solution space. The example with the small solution space allows the optimization to be solved in two ways: using an exhaustive search and using a genetic algorithm. The solution obtained from an exhaustive search is the overall global optimum, whereas the solution from a genetic algorithm is not necessarily the global optimum, but the solutions and the computational demand can be compared. The example with the large solution space is not solvable with the exhaustive approach because the computational demand is too great. Therefore, this example is used to demonstrate the scalability of the genetic algorithm approach.

4.1. Small Solution Space Example

A pressure vessel is subject to structural deterioration due to corrosion. The pressure vessel was inspected several times in the past and was recently inspected again. $J = 4$ defects were detected. From the inspection data, the CDF of the time to failure for each defect can be determined using structural reliability analysis [35]. For the purposes of this example, the CDFs of the failure time for each defect are assumed and provided in Figure 1. The vessel has a remaining service life of 100 years and can be repaired every 5 years. Therefore, there are $m = 20$ potential repair times, and the solution space contains $(m+1)^J = 194{,}481$ candidate solutions.

The repair cost can be assumed as any function of the number of defects k to be repaired. This example defines the repair cost as $C_R(k) = C_R + \omega(k-1)C_R$, where C_R is the cost of the first repair and ω is a factor governing the cost of each additional repair. The cost of repairing zero defects is $C_R(0) = 0$. Only the relative ratio of the costs of failure C_F and first repair C_R are required to populate the objective function, not the absolute costs. The relative expected cost $\mathrm{E}[C]/C_R$ can then be used for the score of the objective function. The values of the input variables should be selected based on all available system information. If these values are uncertain, then this uncertainty can be accounted for by treating the inputs as random variables. The input values assumed in these examples are detailed in Table 1.

For the exhaustive approach, the objective function in (2) is evaluated for each of the 194,481 candidate maintenance plans. The optimal solution is $\mathbf{e} = (60, 15, 60, 40)^{\mathrm{T}}$, with a corresponding objective function score of $\mathrm{E}[C]/C_R = 0.95$. This means that defect 1 should be repaired after 60 years, defect 2 after 15 years, defect 3 after 60 years, and defect 4 after 40 years.

Comparing the optimal repair plan with the CDFs of the time to failure in Figure 1, it can be seen that each defect is repaired before the failure risk increases too drastically. The repair time for defect $j = 3$ would have been later if it was assessed independently, but because defect $j = 1$ was repaired after 60 years, it was less costly to repair defect $j = 3$ at the same time due to the decreased marginal cost of the second repair. The genetic algorithm was able to reach the same optimal solution with a population of 50 members and a crossover rate of 0.8. The progress of the genetic algorithm and exhaustive search methods are illustrated in Figure 2.

Table 1. Input variables for the risk-based maintenance (RBM) analysis using a genetic algorithm.

Variable	Symbol	Value
Failure to repair cost ratio	C_F/C_R	500
Discount rate	r	0.04
Repair factor	ω	0.5

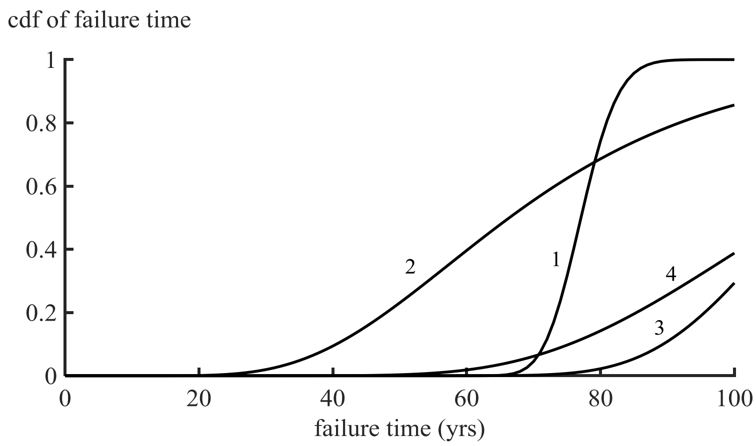

Figure 1. Assumed cumulative distribution functions (CDFs) of times to failure for the four defects. The defect numbers are shown beside the lines.

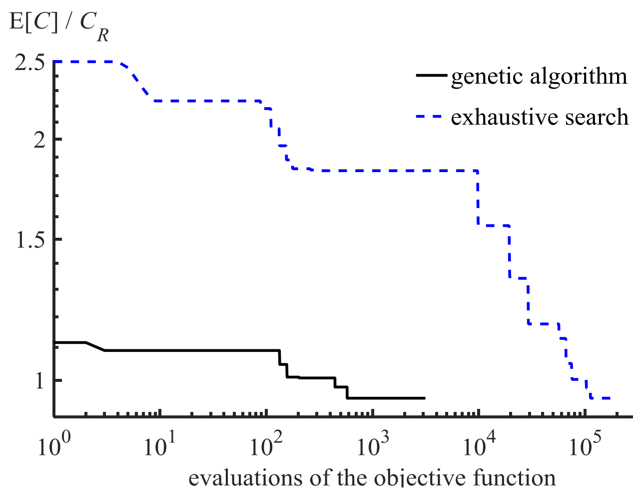

Figure 2. Progress of the genetic algorithm and exhaustive search solutions.

Figure 2 shows the normalized objective function score of the current best solution for each iteration. The genetic algorithm ran for 63 generations, each with a population of 50 members, for a total of 3150 evaluations of the objective function. The globally optimal solution was reached in the

13th generation, after 650 evaluations of the objective function. The algorithm then ran until reaching the termination criteria of 50 generations without improvement in the solution. The computational efficiency of the genetic algorithm can be assessed by comparing it with the exhaustive search method. The exhaustive search evaluated the objective function 194,481 times before it could confirm the optimal solution. Thus, the exhaustive search required 61 times the number of evaluations of the objective function, leading to a factor 7 decrease in computation time for the genetic algorithm. Repeating the genetic algorithm run consistently reveals the globally optimal solution regardless of the values of the initial population.

4.2. Large Solution Space Example

This example considers the same pressure vessel but with a vastly expanded solution space. Instead of four defects, the most recent inspection detected $J = 40$ defects, and instead of the opportunity to repair every 5 years, the vessel can now be repaired annually. Annual repairs for the same 100 year service life yields $m = 100$ potential repair times and $(m+1)^J = 1.5 \times 10^{80}$ candidate solutions. An exhaustive search of 10^{80} candidate solutions is not possible, so a heuristic such as the genetic algorithm is the only way to solve an RBM problem of this scale. The assumed CDFs of the time to failure for each of the 40 defects are shown in Figure 3 (note the CDFs of some defects are beyond 100 years).

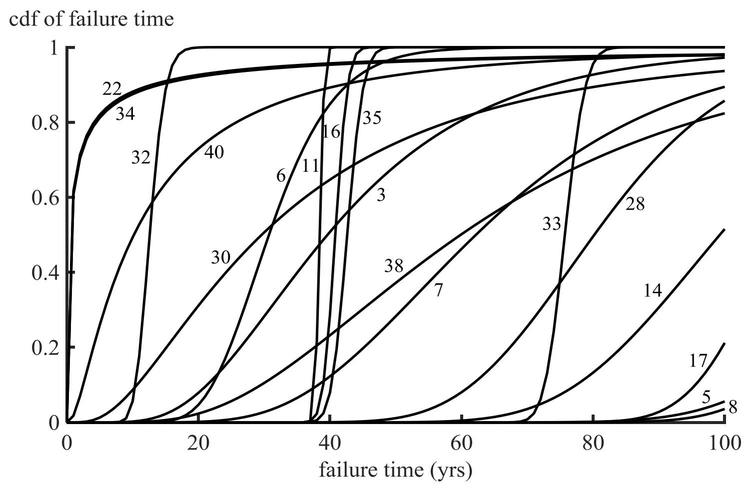

Figure 3. Assumed CDFs of times to failure for the forty defects. The defect numbers are shown beside the lines.

In a problem of this scale, with 40 variables and 10^{80} candidate solutions, it is important to select appropriate parameters for the genetic algorithm. Three main parameters to select are the population size for each generation, the crossover rate, and the stopping criteria. A sensitivity analysis was performed to assess the appropriate ranges of the genetic algorithm parameters for RBI or RBM. The population size and crossover rate were varied, while the stopping criteria was kept constant at 10^5 evaluations of the objective function. A population of 2 times the number of variables ($2J = 80$) consistently produced the best results, along with a crossover rate ranging from 0.6 to 0.9, with 0.9 being the overall best choice. A population size of less than the number of variables (J) was too small to contain enough variety in the initial set, and therefore took many generations to mutate towards the optimal solution. Large population sizes (greater than $4J$) did not require many generations, but were less efficient because the computation time for each generation was slower. As a general rule, this study recommends a population size that is the greater of 2 times the number of optimization variables or 50. However, it is advised to test the performance of the algorithm across a range of population sizes.

The progress of the genetic algorithm is shown in Figure 4. Three runs of the genetic algorithm are shown to demonstrate the impact of crossover and mutation on the solution progress, with extreme crossover rates of 0 and 1, and with an optimal crossover rate of 0.9. All three runs had a fixed population of the optimal $2J$ and a stopping criterion of 10^5 evaluations. At the extremes, with a crossover rate of 0, all children are produced by mutation, and with a crossover rate of 1 all children are produced by crossover. With a crossover rate of 0 the algorithm does not possess any heredity, so the best genetic combinations are not passed on from one generation to the next. The lack of heredity essentially disables the learning aspect of the algorithm, and as can be seen, the algorithm was unable to greatly improve the solution over 10^5 iterations. With a crossover rate of 1 the algorithm does not use any randomly mutated children. Without mutation, the algorithm is at risk of becoming trapped in a local optimum. The objective function scores $\mathrm{E}[C]/C_R$ were 379.7, 6.5, and 6.9 for crossover rates of 0, 0.9, and 1, respectively.

The optimal solution from the run with a crossover rate of 0.9 and a population of $2J$ is shown in Figure 5. To interpret this figure, defect 1 should be repaired after 79 years, defect 2 should be repaired after 97 years, defect 3 should be repaired after 5 years, etc. Again, comparing the repair times for each defect to the failure CDFs in Figure 3 shows the defects are repaired before the failure probability rapidly increases, with a preference towards simultaneous repair. The algorithm was unable to produce a better solution for over 2.2×10^4 evaluations, meaning that the optimal solution seems to be close to the global optimal. As a check, the repair time for each defect can be adjusted up and down by one year from the optimal solution. If this adjustment produces improvements in the optimal solution for many of the defects, then the algorithm is not producing an acceptable solution. In this example, none of the adjusted permutations are improvements, demonstrating the effectiveness of the genetic algorithm.

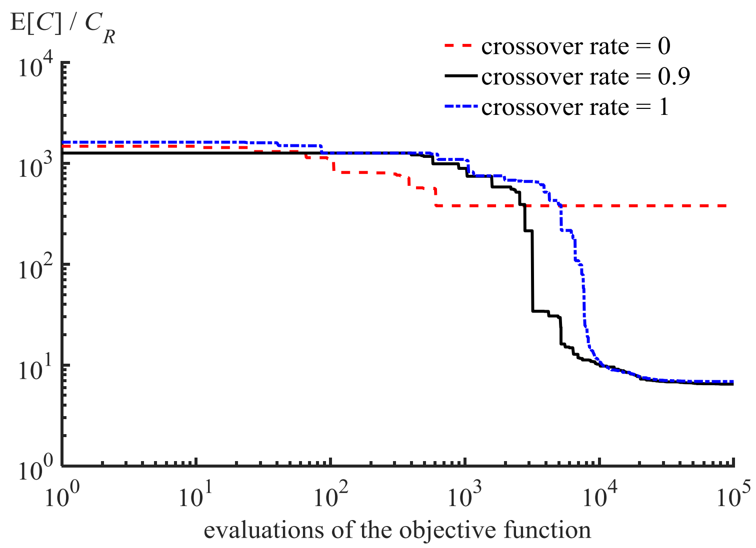

Figure 4. Progress of the genetic algorithm with different crossover rates.

This example demonstrates the value of genetic algorithms in solving the RBI and RBM optimization problem. It is not practical to solve an optimization problem of this scale using the exhaustive search method because the computational requirement of 10^{80} evaluations of the objective function is too great. In contrast, the genetic algorithm reached a solution that approached the global optimum within 10^5 evaluations of the objective function. For reference, the elapsed time for this computation on a standard computer was approximately 8 min, and a more lenient stopping criterion would further reduce this time.

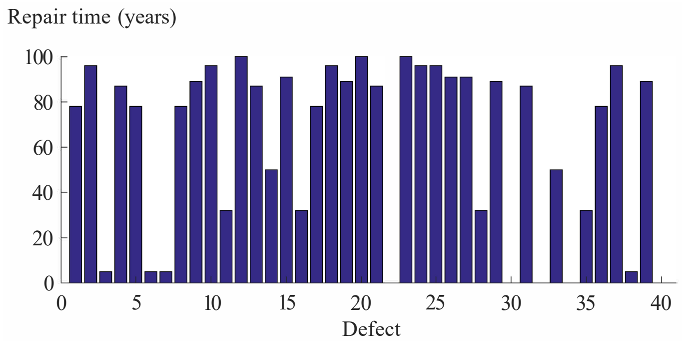

Figure 5. Optimal solution for the case with a crossover rate of 0.9 and a population of $2J$. A repair time of 0 years is the current time, and a repair time of 100 years actually means never, as 100 years is the end of the service life.

5. Conclusions

This paper presents a method for efficiently determining optimal risk-based inspections and maintenance plans using a genetic algorithm. In risk-based inspection and maintenance planning, the optimal plan is the one that maximizes the utility provided by the system over its lifecycle. For simple systems, determining the optimal plan is straightforward. However, for more complex systems, the solution space of all possible inspection and maintenance plans can be very large, and searching the entire solution space is not feasible. This paper demonstrates that genetic algorithms can be successfully used as a heuristic to more efficiently determine the optimal inspection and maintenance plan. Two examples were used to illustrate the method. First, a risk-based maintenance optimization problem for a pressure vessel was presented, with a relatively small solution space of 10^5 candidate solutions. Optimization with a genetic algorithm was compared to an exhaustive search, and it was found that the genetic algorithm yielded the same optimal plan as the exhaustive method but was seven times faster in terms of computation time. Second, the example problem was expanded to entail a much larger solution space of 10^{80} candidate solutions. This problem was too large to solve with an exhaustive search, but the genetic algorithm was still able to determine a solution with a relatively short computation time on a standard computer. One limitation of the application of this method to more complex systems is that it cannot be known whether the optimal solution determined is the global optimum or just a local optimum. However, the alternative is to simplify the problem, which can lead to unrealistic results. The larger optimization problem is more realistic in scale to a problem that could be faced by the operator of an engineering system. Some of challenges in applying this methodology are in developing functions to describe the time to failure of the system from reliability theory, determining the dependency of the variables in the objective function in (1) on the inspection and maintenance plan, and developing the equations to describe this dependency. Genetic algorithms are relatively simple to implement, and are supported by many software packages (the analysis in this study was performed with Matlab). Thus, this paper shows that genetic algorithms are a practical and effective method of solving risk-based inspection and maintenance planning problems for real-world engineering systems. Future work in this area should explore alternative solutions to more efficiently solve complex risk-based inspection problems such as other heuristic algorithms or simplifications to the solution space that can still yield accurate results.

Author Contributions: Conceptualization, M.R.D.; Methodology, S.H.; Software, S.H.; Validation, M.R.D.; Formal Analysis, S.H.; Investigation, S.H.; Resources, M.R.D.; Data Curation, S.H.; Writing—Original Draft Preparation, S.H.; Writing—Review & Editing, M.R.D.; Visualization, S.H.; Supervision, M.R.D.; Project Administration, M.R.D.; Funding Acquisition, M.R.D. and S.H.

Funding: This research was funded by NSERC Grant number RGPIN-2015-04135. The first author is also grateful for support from the NSERC PGS-D scholarship, the Queen Elizabeth II Graduate Scholarship, and the Mo Mohitpour Graduate Scholarship in Pipeline Engineering.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CDF	cumulative distribution function
RBI	risk-based inspection
RBM	risk-based maintenance

References

1. Rackwitz, R.; Lentz, A.; Faber, M. Socio-economically sustainable civil engineering infrastructures by optimization. *Struct. Saf.* **2005**, *27*, 187–229. [CrossRef]
2. Von Neumann, J.; Morgenstern, O. *Theory of Games and Economical Behaviour*, 2nd ed.; Princeton University Press: Princeton, NJ, USA, 2007.
3. Luce, R.D.; Raiffa, H. *Games and Decisions: Introduction and Critical Survey*; John Wiley and Sons: New York, NY, USA, 2012.
4. Jordaan, I. *Decisions under Uncertainty: Probabilistic Analysis for Engineering Decisions*; Cambridge University Press: Cambridge, UK, 2005; ISBN 978-0-521-78277-5.
5. Parmigiani, G.; Inoue, L. *Decision Theory: Principles and Approaches*; John Wiley & Sons Ltd.: Chichester, UK, 2009; ISBN 978-0-471-49657-1.
6. Pratt, J.W.; Raiffa, H.; Schlaifer, R. *Introduction to Statistical Decision Theory*; MIT Press: Cambridge, MA, USA, 1995; ISBN 978-0-262-16144-2.
7. Straub, D. Generic Approaches to Risk Based Inspection Planning for Steel Structures. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 2004.
8. Khan, F.I.; Haddara, M.M. Risk-based maintenance (RBM): A quantitative approach for maintenance/inspection scheduling and planning. *J. Loss Prev. Process Ind.* **2003**, *16*, 561–573. [CrossRef]
9. Faber, M.H.; Straub, D.; Goyet, J. Unified approach to risk-based inspection planning for offshore production facilities. *J. Offshore Mech. Arct. Eng.* **2003**, *125*, 126–131. [CrossRef]
10. Fujimoto, Y.; Kim, S.C.; Hamada, K. Inspection Planning of Fatigue Deteriorating Structures Using Genetic Algorithm. *J. Soc. Naval Archit.* **1997**, *182*, 729–739. [CrossRef]
11. Goyet, J.; Straub, D.; Faber, M.H. Risk-based Inspection Planning of Offshore Installations. *Struct. Eng. Int.* **2002**, *12*, 200–208. [CrossRef]
12. Garbatov, Y.; Soares, C.G. Cost and reliability based strategies for fatigue maintenance planning of floating structures. *Reliab. Eng. Syst. Saf.* **2001**, *73*, 293–301. [CrossRef]
13. Pandey, M.D. Probabilistic models for condition assessment of oil and gas pipelines. *NDT E Int.* **1998**, *31*, 349–358. [CrossRef]
14. Hellevik, S.G.; Langen, I.; Sørensen, J.D. Cost optimal reliability based inspection and replacement planning of piping subjected to CO_2 corrosion. *Int. J. Press. Vessels Pip.* **1999**, *76*, 527–538. [CrossRef]
15. Sahraoui, Y.; Khelif, R.; Chateauneuf, A. Maintenance planning under imperfect inspections of corroded pipelines. *Int. J. Press. Vessels Pip.* **2013**, *104*, 76–82. [CrossRef]
16. Gomes, W.J.S.; Beck, A.T.; Haukaas, T. Optimal inspection planning for onshore pipelines subject to external corrosion. *Reliab. Eng. Syst. Saf.* **2013**, *118*, 18–27. [CrossRef]
17. Zhang, S.; Zhou, W. Cost-based optimal maintenance decisions for corroding natural gas pipelines based on stochastic degradation models. *Eng. Struct.* **2014**, *74*, 74–85. [CrossRef]
18. Chang, M.-K.; Chang, R.-R.; Shu, C.-M.; Lin, K.-N. Application of risk based inspection in refinery and processing piping. *J. Loss Prev. Process Ind.* **2005**, *18*, 397–402. [CrossRef]
19. Stewart, M.G. Reliability-based assessment of ageing bridges using risk ranking and life cycle cost decision analyses. *Reliab. Eng. Syst. Saf.* **2001**, *74*, 263–273. [CrossRef]
20. Barone, G.; Frangopol, D.M. Life-cycle maintenance of deteriorating structures by multi-objective optimization involving reliability, risk, availability, hazard and cost. *Struct. Saf.* **2014**, *48*, 40–50. [CrossRef]

21. Martorell, S.; Carlos, S.; Sánchez, A.; Serradell, V. Constrained optimization of test intervals using a steady-state genetic algorithm. *Reliab. Eng. Syst. Saf.* **2000**, *67*, 215–232. [CrossRef]
22. Marseguerra, M.; Zio, E. Optimizing maintenance and repair policies via a combination of genetic algorithms and Monte Carlo simulation. *Reliab. Eng. Syst. Saf.* **2000**, *68*, 69–83. [CrossRef]
23. Straub, D.; Faber, M.H. Risk based inspection planning for structural systems. *Struct. Saf.* **2005**, *27*, 335–355. [CrossRef]
24. Joint Committee on Structural Safety. Risk Assessment in Engineering; Principles, System Representation & Risk Criteria. Available online: http://www.jcss.byg.dtu.dk/ (accessed on 25 April 2018).
25. Gomes, W.J.S.; Beck, A.T. Optimal inspection and design of onshore pipelines under external corrosion process. *Struct. Saf.* **2014**, *47*, 48–58. [CrossRef]
26. Hong, H.P. Reliability based optimal inspection and maintenance for pipeline under corrosion. *Civ. Eng. Syst.* **1997**, *14*, 313–334. [CrossRef]
27. Hong, H.P. Inspection and maintenance planning of pipeline under external corrosion considering generation of new defects. *Struct. Saf.* **1999**, *21*, 203–222. [CrossRef]
28. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley Professional: Boston, MA, USA, 1989; ISBN 978-0201157673.
29. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press/Bradford Book: Cambridge, MA, USA, 1992; ISBN 978-0262581110.
30. Kirkpatrick, S.; Jelatt, C.C.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–769. [CrossRef]
31. Xu, Z. *Uncertain Multi-Attribute Decision Making: Methods and Applications*; Springer: Berlin, Germany, 2015; ISBN 978-3662456392.
32. Sierksma, G.; Zwols, Y. *Linear and Integer Optimization: Theory and Practice*, 3rd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2015; ISBN 978-1498710169.
33. Affenzeller, M.; Wagner, S.; Winkler, S.; Beham, A. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2009; ISBN 978-1584886297.
34. Deep, K.; Singh, K.P.; Kansal, M.L.; Mohan, C. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Appl. Math. Comput.* **2009**, *212*, 505–518. [CrossRef]
35. Melchers, R.E. *Structural Reliability Analysis and Prediction*; John Wiley & Sons Ltd.: Chichester, UK, 2018; ISBN 0-471-98324-1.