PAVEMENT MAINTENANCE DECISION OPTIMIZATION USING A NOVEL DISCRETE BARE-BONES PARTICLE SWARM ALGORITHM

M. Sh. Mahmood[1]
Ph.D. researcher
School of Architecture, Design and the Built Environment
Nottingham Trent University, Nottingham, NG1 4BU, United Kingdom
Phone, Fax: +44 7428790585
E-mail: maher.mahmood2010@my.ntu.ac.uk

S. Mathavan
Visiting Research Fellow
School of Architecture, Design and the Built Environment
Nottingham Trent University, Nottingham, NG1 4BU, United Kingdom
E-mail: s.mathavan@ieee.org

M. M. Rahman
Senior Lecturer
Department of Mechanical, Aerospace and Civil Engineering
Brunel University, Uxbridge, UB8 3PH, United Kingdom
E-mail: mujib.rahman@brunel.ac.uk

ABSTRACT
Timely pavement maintenance and rehabilitation is essential for a healthy road network. As the resources are always limited, some form of action prioritization is necessary. There are a number of objectives to satisfy and the influencing variables are too many, leading to complicated decision making scenarios. In this work, a novel bare-bones particle swarm algorithm is presented for a general multi-objective problem that is discrete in nature. In contrast to the original particle swarm method, the proposed technique has the advantage in that it is a parameter-free technique. The developed algorithm is applied to find optimal rehabilitation scheduling considering the two objectives, the minimization of the total pavement rehabilitation cost and the minimization of the sum of all residual pavement condition index (PCI) values. The method is benchmarked against a discrete-domain particle swarm algorithm, by comparing a number of performance criteria, demonstrating its effectiveness.
Keywords: pavement maintenance, pavement management, multi-objective optimization, particle swarm, bare-bones.

[1]Corresponding author
Submission date: 14/07/2016
Word count: 7363 words, 10 Figures, 5 Tables

INTRODUCTION
Highways play an important role in the economic and social well-being of a country at the national and local levels. Pavement is a key element of road infrastructure. Increasing traffic volumes, heavier loads and poor reinstatement following excavation by public utility companies allied with repeated adverse weather conditions are causing significant functional and structural deterioration in the pavement such as cracking, localized depression, rutting, potholes, texture loss, etc. Increasing demands to repair, associated with increased pavement deterioration, as well as deficient resource allocation, have made the task of maintaining pavement network more challenging and difficult (*1*). Regular maintenance and rehabilitation (M&R) is essential to preserve and improve a pavement network. Because of limited availability of resources, maintenance activities must be timely and effective. Unnecessary maintenance increases overall maintenance costs, whereas delayed maintenance may increase rehabilitation costs. In

recent years, therefore, efficiency has become a key issue in highway pavement maintenance planning (*2*).

Pavement management systems (PMSs) are becoming progressively essential tools in the decision-making procedures regarding the preservation of pavement networks. A perfect PMS is a program that would keep all pavement segments at satisfactorily high serviceability and structural conditions. At both network level and project level, many highway agencies employ prioritization programming models to compare pavement investment alternatives. The majority of highway authorities in the world have use different PMS computer programs such as PAVER, HDM-4, UKPMS. These programs are developed by using decision trees or heuristics (*3*). In prioritization models, the pavement condition data are used to find a factor or index to represent the present pavement condition. Prioritization is done by ranking all the pavement segments based on a priority-ranking index. This ranking index usually considers different parameters such as highway class, traffic volume, quality index, etc. The maintenance and rehabilitation needs selection and budget allocation are often conducted based on this priority-ranking index (*4*, *5*).

An alternative approach to prioritization, in the form of optimization, is also used. A PMS is required to keep all pavement segments at satisfactorily high serviceability and structural conditions. However, it shall only require minimum resources (budget, equipment, manpower, etc.) and should not produce any significant negative effect on the environment, safe traffic operations, and social and community activities. Since many of these objectives are conflicting requirements, the decision-making process of PMSs for scheduling pavement maintenance activities should involve a multi-objective consideration that handles the competing requirements of different objectives (*6*). Optimization has been widely adopted for selecting pavement maintenance plans. In this regard, many mathematical programming techniques (e.g. linear and dynamic programming), computational intelligence methods (e.g. genetic algorithms and particle swarms) or hybrid models that combine the two techniques have been used (*6*). The mathematical programming techniques are limited application and designed for particular optimisation problems.

Many researchers use genetic algorithm for single and multi-objective optimization for pavement decision making problems {Single objective GA (*7–20*) Multi objective GA (*6*, *11*, *21–24*)}. When it comes to using particle swarm optimization (PSO) for pavement problems. Wang and Goldschmidt (2008) proposed a project interaction pre-optimization model that integrates the project interaction, traffic-demand prediction interaction and maintenance-condition interaction into the decision optimization process. The pre-optimization model was used as an input of a global multi-objective optimization model-based PSO. The multi-objective PSO problem was converted into a single-objective problem by using the weighted aggregation method (*25*). Shen et al. (2009) used chaos PSO (CPSO), a new random global optimization algorithm which has strong local searching capability, in their pavement maintenance decision programming. It was applied on an expressway network to satisfy just a single objective, which was maximization of economic benefit. The pavement maintenance decision results proposed by the CPSO were validated by comparing with the results of the NSGA-II algorithm. It was found that the convergence speed of CPSO to reach the optimal solution was quicker than the convergence speed of NSGA-II (*26*). In 2010, Tayebi and Hassani used PSO with single-objective function scenarios for a pavement management system at the network level. The same hypothetical problem formulation of the Pavenet_R model by (*10*) was used to apply a PSO algorithm for pavement maintenance programming (*27*). Chou and Le (2011) formulated a multi-objective PSO algorithm (i.e. classical one) to study the effect of overlay maintenance activities on the performance pavement reliability with an optimized treatment cost. The maintenance cost and performance reliability of the pavement were considered simultaneously in the developed algorithm as multi-objective functions. For considering uncertainties of input parameters and maintenance effect on pavement service life, a probabilistic model integrated with a Monte Carlo simulation was proposed to predict performance reliability (*28*). Since the genetic algorithm and PSO involve many parameters (such as mutation operator, crossover operator, mutation probability, crossover probability and population size), that require the user to choose a number of parameters. Furthermore, the final performance of these algorithms depends on the value chosen by the user, making their use more difficult for inexperienced people. This difficulty highlights the need for a parameter-free algorithm and this paper presents one.

DESCRIPTION OF THE PAVEMENT MAINTENANCE DECISION PROBLEM

Optimization Problem Parameters

The M&R analysis procedure depends on the following data and decision criteria: current state of the pavement based on distresses, minimum acceptable serviceability level, treatment cost and budget, and analysis period. For determining the treatment needs, the highway network is divided into a number of pavement segments of predefined length (*4*, *10*).

Agency cost of highway asset is the intervention cost which is necessary to design, construct, and maintain a highway network. It consists of highway maintenance, rehabilitation and reconstruction cost. Rehabilitation is necessary for the highway asset at least one time over its lifetime to keep it above the minimum acceptable serviceability and safety level. The cost of any particular rehabilitation activity, which is a form of construction, comes from: materials, preliminary engineering, and construction management (*29*). If a rehabilitation action is to be applied in subsequent years, then the costs of it can be discounted to present worth in the following manner:

$$\text{Present cost} = \text{Future cost} \times \text{PWF} \quad (1)$$

where PWF is the present worth factor, given by:

$$\text{PWF} = \frac{1}{(1+R)^t} \quad (2)$$

The typical range of discount rates R recommended by FHWA is 3 to 5% (*30*), t = time at which the money is spent (specified in years).

Depending on the situation, highway agencies have the option to choose a rehabilitation action from a list of activities. One such list, which is also used in this work, is given in Table 1. It is also essential to specify the trigger value for each treatment action. A warning level is defined as the minimum level of pavement performance, such that the treatment must be applied when the pavement reaches it. The total span of the analysis period is commonly specified by the highway authority concerned. Furthermore, the length of unit planning period, which is commonly one year, is selected depending on the requirements of the highway authority (10).

TABLE 1 The maintenance and Rehabilitation (M&R) Strategies

No.	M&R strategy
1	Do nothing
2	AC* overlay 1in (25mm)
3	AC overlay 2in (50mm)
4	AC overlay 4in (100mm)
5	AC overlay 6in (150mm)

* Asphalt Concrete

Objective Functions

The common objectives of pavement maintenance systems as identified by road authorities comprise the following: to minimize the present worth of overall treatment costs over the analysis period; to minimize user costs by choosing and scheduling treatment actions to decrease delays and disruptions to traffic; and to keep the serviceability of the pavement network over the minimum acceptable level with the resources available. Commonly, two or more of these objectives are combined by allocating a proper weighting factor to each (*10*).

The main challenge in pavement management is the selection of maintenance investment alternatives for a large number of pavement sections over multiple time periods (*31*). To reach the optimal maintenance investment decisions, it is important to optimize the M&R decision considering multiple objectives such as minimum cost and maximum performance, etc. To address complex optimization problem of pavement management, multi-objective programming of pavement management activities is developed using the particle swarm optimization technique.

The multi-objective programming of pavement management can be presented mathematically as follows:

Minimize the total pavement maintenance cost

$$f_1(x) = \sum_{t=1}^{T} \sum_{p=1}^{N} \sum_{m=1}^{M} x_{m,p,t}\, C_m\, L_p\, W_p\, (1+R)^{-t} \quad (3)$$

and minimize the sum of all residual PCI values to maximize the PCI of candidate section.

$$f_2(x) = \sum_{t=1}^{T} \sum_{p=1}^{N} \sum_{m=1}^{M} x_{m,p,t} \left[\left(\text{PCI}_{max} - \text{PCI}_{p,t} \right) L_i \, W_p \, \text{AADT}_{p,t} \right] \quad (4)$$

where $x_{m,p,t} = \begin{bmatrix} 1 & \text{if treatment } m \text{ for section } p \text{ at time } t \text{ is selected} \\ 0 & \text{otherwise} \end{bmatrix}$

In the equations above, m is the treatment type; M stands for the total number of different treatment types; p is the pavement section number under consideration; N is the total number of pavement sections; t is any time in the analysis period, and T is the total analysis period (both are usually specified in years); C_m is the unit cost of treatment type m; L_p is the length of pavement section p; W_p stands for the width of section p; R is the discount rate; $\text{PCI}_{p,t}$ = PCI for section p at time t; PCI_{max} is the maximum PCI level (100 %); $\text{AADT}_{p,t}$ is the annual average daily traffic for section p at time t. In this work, the following acceptable level for section performance is chosen: $\text{PCI}_{p,t} \geq 65$ %.

Pavement Deterioration Model

A PMS must predict the performance of a pavement network for the subsequent years in order to evaluate the outcome of a given set of maintenance decisions, thereby enabling it to optimize the maintenance decision. A pavement deterioration model is an essential component when determining treatment needs, and when estimating highway user costs and benefits of the treatment application (*32*). In general, deterioration models are established in terms of a pavement condition indicator and the exogenous influences contributing to pavement deterioration (*22*). Various researchers have developed network-level deterministic deterioration prediction models for flexible pavements, to predict pavement deterioration by considering distress, pavement age, traffic loading, and maintenance effects. Here, a deterministic deterioration model for arterial highways in the wet freeze climatic region has been designed to estimate future pavement condition, described, in detail, in the previous work of the authors (*33*) :

$$PCI = 97.744 - 0.15\, cracking\ area\ (alligator, edge, and\ block) - 0.064\, total\ longitudinal\ and\ transverse\ cracking\ length - 0.515\, pavement\ age + 3.748\, maintenance\ effect\ (inlay\ and\ overlay\ thickness) \quad (5)$$

where PCI is the pavement condition index. It should be noted that this optimization method is dependent of any particular deterioration prediction model.

PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a simulation of the social behavior of birds or fish within their flock or school, and was developed by Kennedy and Eberhart in 1995 (*34*). The swarm of PSO comprises a set of particles, each particle representing a possible solution of an optimization problem. Each particle moves in the search space, and this movement is achieved by the operator that is directed by a local element and by social elements. Each solution or particle is assumed to have a position and a velocity. The position and velocity of the ith particle is denoted at iteration z by $X_i(z) = \{X_{i,1}(z), X_{i,2}(z), \ldots, X_{i,n}(z)\}$ and $V_i(z) = \{ V_{i,1}(z), V_{i,2}(z), \ldots, V_{i,n}(z)\}$. Here, n is the dimension of the search space, where n = N×T. Then, each particle i updates the position and velocity of its jth dimension at iteration z + 1 by using the following equations (*35*, *36*):

$$V_{i,j}(z+1) = w\, V_{i,j}(z) + r_1\, c_1 \left[Pbest_{i,j}(z) - X_{i,j}(z) \right] + r_2\, c_2 \left[Gbest(z) - X_{i,j}(z) \right] \quad (6)$$

$$X_{i,j}(z+1) = X_{i,j}(z) + V_{i,j}(z+1) \quad (7)$$

where $Pbest_{i,j}(z)$ is the local or personal best position for the jth dimension of particle i at iteration z; $Gbest(z)$ is the global best position or particle leader at iteration z; w is the inertia weight of the particle; c_1 and c_2 are acceleration coefficients that are positive constants; r_1 and r_2 are random numbers in [0,1].

In the velocity update equation, the leader particle Gbest in each generation guides the particles to move towards the optimal positions. In each generation, the particle memory is updated. For each particle in the swarm, performance is estimated according to the fitness function or objective function of the optimization problem. The inertia weight w is used to regulate the effect of the previous velocities on the current velocity, and hence to effect a trade-off between the global and local exploration abilities of the particles (*37*).

Multi-Objective Optimization Problems

Multi-objective optimization problems include the simultaneous satisfaction of two or more objective functions. Furthermore, the multiple objectives of optimization problems are usually conflicting objectives, which means there is no single optimal solution. Therefore, it is necessary to find a decent trade-off of solutions that represent a compromise between the objectives. In multi-objective particle swarm optimization (MOPSO) problems, the main challenge is to determine the best global particle "leader" at each generation. In a single-objective problem, the leader particle is found easily by choosing the particle that has the best position. For multi-objective problems there is a set of non-dominated solutions called "Pareto-optimal solutions", which is the set of best solutions (*37*).

The feasible solutions of a multi-objective optimization problem are Pareto-optimal solutions if there are no other feasible solutions that can yield progress in one objective without damaging at least one other objective (*38*). The Pareto optimality is named after Vilfredo Pareto. The definition of Pareto optimality is that "A decision vector, $\mathbf{x}^* \in \mathcal{F}$, is Pareto-optimal if there does not exist a decision vector, $\mathbf{x} \neq \mathbf{x}^* \in \mathcal{F}$ that dominates it. For maximization problems, this condition can be expressed as, $\nexists k : f_k(\mathbf{x}) < f_k(\mathbf{x}^*)$. For minimization problems, $\mathbf{x}^* \in \mathcal{F}$ will be Pareto-optimal if $f_k(\mathbf{x}) > f_k(\mathbf{x}^*)$ for any $\mathbf{x} \neq \mathbf{x}^* \in \mathcal{F}$. An objective vector, $\mathbf{f}^*(\mathbf{x})$, is Pareto optimal if x is Pareto optimal" (*39*). For a set of objective functions $\{f_1, f_2, \ldots, f_K\}$, the condition that a feasible solution $\mathbf{x}^*$ dominates another feasible solution x, then it is denoted by $\vec{F}(\mathbf{x}^*) \prec \vec{F}(\mathbf{x})$, the target being maximization.

Discrete (Binary) Particle Swarm Optimization

The most common optimization problems have either discrete or qualitative distinctions between variables. In the discrete PSO, the solutions can be assumed to be one of the several discrete values. The most common example of a discrete PSO is binary optimization, where all solutions will be 0 or 1. Fundamentally, the continuous domain PSO is different from a discrete PSO in two ways. Firstly, the particle coordinate is composed of binary values. Secondly, the velocity must be transformed into a probability change, that is, the chance of the binary variable taking the value of 1 (*40*, *41*).

The algorithm of PSO for continuous optimization problems was modified for solving discrete (binary) optimization problems by changing the position equation to a new one. The following is an equation for the modified algorithm (*40–42*):

$$X_{i,j} = \begin{cases} 1 & \text{if } rand() < S(V_{i,j}) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $rand()$ is a quasi-random number chosen from the continuous uniform distribution in the interval [0,1], i.e. $U[0,1]$, and $S(V_{i,j})$ is the sigmoid function given by

$$S(V_{i,j}) = \frac{1}{1+e^{-X_{i,j}}} \tag{9}$$

Barebones Particle Swarm Optimization (BBPSO)

The behavior of a particle is such that it converges to a weighted average between its local best position and the global best position. This behavior induced Kennedy to modify the original algorithm by replacing the equation of the particle velocity with a Gaussian sampling based on $Pbest_i(z)$ and $Gbest(z)$, resulting in BBPSO. The velocity equation of the original algorithm is replaced by (*39*, *43*):

$$X_{i,j}(z+1) = N\left(\frac{Pbest_{i,j}(z)+Gbest(z)}{2}, \left|Pbest_{i,j}(z) - Gbest(z)\right|\right) \tag{10}$$

Where, N denotes a Gaussian distribution. Based on this equation, the particle position is randomly chosen from the Gaussian distribution with the mean of the local best position and the global best position. In addition, Kennedy developed another version of the BBPSO, symbolized by BBExp, by modifying the equation thus:

$$X_{i,j}(z+1) = \begin{cases} N\left(\frac{Pbest_{i,j}(z)+Gbest(z)}{2}, \left|Pbest_{i,j}(z) - Gbest(z)\right|\right) & if\ U(0,1) < 0.5, \\ Pbest_{i,j}(z) & otherwise \end{cases} \tag{11}$$

As there is a probability of 50% that the jth dimension of a particle changes to the corresponding local best position, the new version of the algorithm tends to search for local best positions. The main strengths of BBPSO are that it is parameter-free and appropriate for application to real problems where the information on inertia weights and acceleration coefficients of particles is insufficient or difficult to obtain (*43*). In addition, it is easy to implement and performs well when dealing with multi-objective optimization problems (*43*).

DISCRETE BAREBONES MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION (DBB-MOPSO)

In this section a discrete version of the BBPSO, called discrete multi-objective PSO (DBB-MOPSO), is proposed for multi-objective optimization problems. The process flow of the DBB-MOPSO algorithm is shown in Figure 1. The process stages are as follows.

Initialization

Particle Positions

The first step in the initialization stage of DBB-MOPSO is randomly generating the swarm with a predefined size. For each particle, values are assigned for each dimension randomly from a predefined set of values, as explained in detail below (*43*).

One of the main steps in designing an effective particle swarm optimization algorithm is the correct representation of particle positions for finding a proper mapping between the problem solution and the particle. There are two forms of representation, namely direct and indirect representations (*44*). In this research, a combination of direct and indirect representation is adopted. A problem solution (position) in direct representation is encoded in a one dimensional string of size n, where n = N×T. Every element of the string is a number chosen randomly from the set {1,2,3,….,M}, where for the problem at hand, M is the number of pavement maintenance actions. For the current problem, the structure of direct encoding is shown in Figure 2:

In indirect encoding, solutions for each particle are encoded in a position matrix, n×M. In the position matrix, the values of the matrix elements for each particle are binary values, 0 or 1. Moreover, in each column the value of most of the elements is 0; just one element, corresponding to the maintenance action, is 1. For the direct representation in Figure 2, the indirect encoding is shown in Figure 3:

Particle Velocity, Local Best Position

Indirect encoding is used to initialize the velocity of each particle. The n×M matrix is generated and all elements of the matrix are assumed to be 0. The initial personal best position of each particle is assumed to be equal to the initial position of the particle, $Pbest_{i,j}(0) = X_{i,j}(0)$, where $X_{i,j}(0)$ is the initial position of the jth dimension of the ith particle in the swarm. To save the non-dominated solutions found across all iterations, an archive, or memory, is initialized from the initial swarm.

Updating The Local Best Positions

The local best position for particle i, $Pbest_i(z)$, is the best position reached by the particle itself to date. The local best position is updated at each iteration according to the equation (12). If the fitness value of the previous $Pbest_i(z)$ is smaller than the fitness value of the current position $X_i(z+1)$, the current $Pbest_i(z)$ will not be replaced. Otherwise, it will be replaced by the current position $X_i(z+1)$ (*43*).

$$Pbest_i(z+1) = \begin{cases} Pbest_i(z), & \text{if } \vec{F}(Pbest_i(z)) \prec \vec{F}(X_i(z+1)) \\ X_i(z+1), & \text{otherwise} \end{cases} \quad (12)$$

where i = 1, 2, …, I, and I is the total number of particles in the swarm (i.e. the swarm's size).

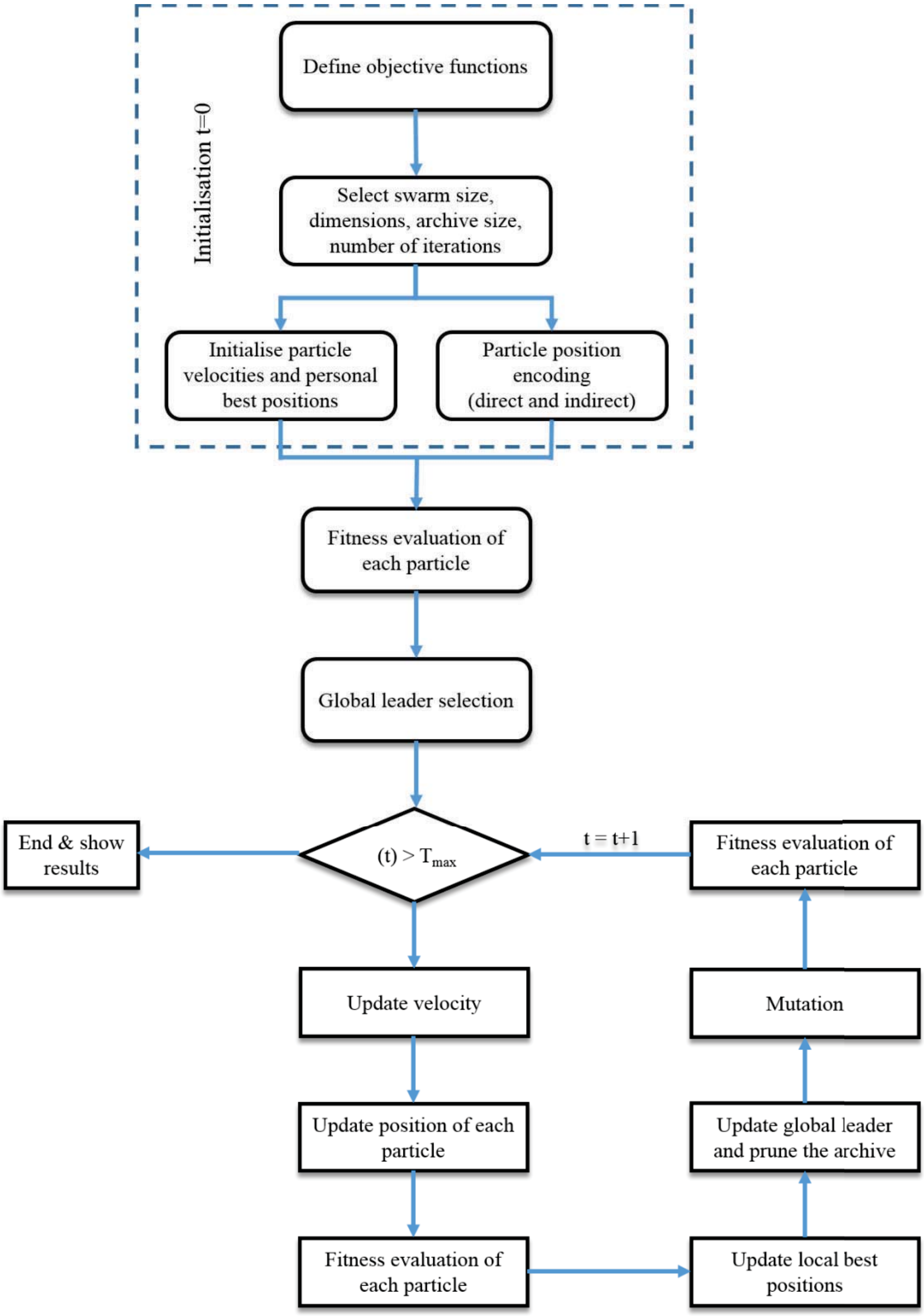

FIGURE 1 Flow chart of the binary barebones particle swarm optimization algorithm

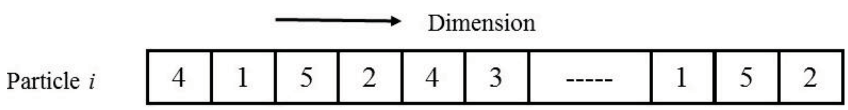

FIGURE 2 Direct representation (encoding)

Maintenance action 1	0	1	0	0	0	0	-----	0	0	0
Maintenance action 2	0	0	0	1	0	0	-----	0	0	1
Maintenance action 3	0	0	0	0	0	1	-----	0	0	0
Maintenance action 4	1	0	0	0	1	0	-----	0	0	0
Maintenance action 5	0	0	1	0	0	0	-----	1	1	0

FIGURE 3 Indirect representation (encoding) for particle *i*

Updating the Global Best Positions

The leader particle or global best position $Gbest(z)$ is the best solution found from the swarm of particle neighbours so far. For single-objective optimization problems the global best position is found in a straightforward manner. Conversely, in multi-objective optimization problems, the multiple conflicting objectives make it challenging to select a leader solution. To overcome this problem, DBB-MOPSO is designed to maintain a memory (archive) with a sufficient capacity to store the non-dominated (Pareto) solutions, as proposed by (*39*, *43*).

To find the leader particle, the sigma method is used here. This method was developed by Mostaghim and Teich (2003). In this method, a value σ_i is assigned to each solution with coordinates $\left(f_{1,i}, f_{2,i}\right)$, and thus all the solutions that are on the line $(f_1 = \sigma f_2)$ have the same σ value. The sigma value (σ) can be determined for two objectives as follows:

$$\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2} \tag{13}$$

For more than two objective functions, Mostaghim and Teich [2003] provide the formulae for the estimation of σ. The leader particle $Gbest(z)$ among the archive members of each generation is selected as follows. Firstly, the sigma value σ is assigned to each non-dominated solution e in the archive. Secondly, the sigma value is determined for particle a of the current generation. Then, the distance between them (σ_e, σ_a) is calculated. Finally, solution g in the archive that has the lowest distance to solution a is chosen as the global best position or leader particle. Therefore, each solution which has a closer sigma value to the sigma value of a non-dominated solution must choose that non-dominated solution as the leader solution (*45*).

Updating Particle Velocities and Positions

To handle the multi-objective optimization problem, a new version of BBExp, namely BBVar, has been proposed to update a particle's position by (*43*), and it works as shown below:

$$X_{i,j}(z+1) = \begin{cases} N\left(\frac{r_3\, Pbest_{i,j}(z) + (1-r_3)\, Gbest(z)}{2}, \left|Pbest_{i,j}(z) - Gbest(z)\right|\right), & \text{if } U(0,1) < 0.5, \\ Gbest(z), & \text{otherwise} \end{cases} \tag{14}$$

where, r_3 is a random number chosen from U[0,1]. This formulation avoids the use of particle velocities used in the regular PSO algorithm.

For discrete problems the definition in Equation (14) is of not much use as the resulting positions, for each dimension of a particle, will have to be either 0 or 1. In this work, the velocity term is reintroduced for the discrete barebones algorithm. However, rather than using the parameters as defined in Equation (6), it is proposed to make use of Equation (14), where the difference between the current particle position and the estimated position in the next iteration, by using Equation (14), is defined as the equivalent velocity of the particle. Hence, it is proposed here to make the change in the following manner to update a particle's velocity, to deal with discrete multi-objective problems:

$$V_{i,j}(z+1) = \begin{cases} N\left(\frac{Pbest_{i,j}(z) + Gbest(z)}{2}, \left|Pbest_{i,j}(z) - Gbest(z)\right|\right) - X_{i,j}(z), & \text{if } U(0,1) < 0.5 \\ Gbest(z) - X_{i,j}(z), & \text{otherwise} \end{cases} \tag{15}$$

After (*44*), the particle's position is proposed to be updated as follows:

$$X_{i,j}(z+1)=\begin{cases}1 & \text{if } V_{i,j}(z+1)=max\{V_{i,j}(z+1)\},\ \forall j\in\{1,2,\dots,n\}\\ 0 & \text{otherwise}\end{cases} \tag{16}$$

For particle i, the values of all elements, except one, in each column j of the position matrix are 0, and only the element that has the maximum velocity is assigned 1. If, in a given column, there is more than one element with the maximum velocity value, then one of these elements is assigned 1 randomly (*44*). The same method is used by the DBB_MOPSO algorithm presented here.

Mutation Operator

The main feature of PSO is the fast speed of convergence. However, in multi-objective optimization, the PSO algorithm could converge to non-optimal solutions. To prevent a premature convergence to non-optimal solutions in the MOPSO, a mutation operator is used to control convergence speed. In addition, it allows the MOPSO algorithm to expand the search capability, thus gaining better diversity. At the beginning of the generation process, all particles of the swarm are affected by the mutation operator with the full range of decision variables, with the influence of the mutation operator declining as the iteration number increases (*43*). The procedure of mutation operation is given by the following pseudo-code:

```
FUNCTION MUTATION: Out = MUTATE (X, Z)   //X = any particle in the swarm; Z = max. no.
                                           of iterations//
FOR i = 1 TO I                            // For all the particles //
      IF e^((-8*z)/Z) > r_4               // r_4 is a random number chosen from U[0,1] //
            FOR j = 1 TO n                // Do it for all the dimensions //
                  Position X_i,j == 1     // j number chosen randomly from the set
                                            {1,2,3,….,M}
                                            randomly //
            END FOR
      END IF
End FOR                                   // Return the swarm after mutation //
```

External Archive Pruning

In multi-objective optimization algorithms, it is necessary to retain the non-dominated solutions generated across all iterations of the search. In each generation, all new non-dominated solutions are stored in the external archive, while all solutions which became dominated are eliminated. It is common to adopt an external archive with limited capacity characteristics (*43*, *46*). To avoid reaching the maximal capacity of the external archive, crowding distance is used to eliminate some solutions without a negative effect on its distribution. When the archive capacity has reached the maximum limit, the solutions that have the largest crowding distance values are retained in the archive (*43*). The following pseudo-code is the pruning archive procedure.

```
FUNCTION PRUNING ARCHIVE: Opt = PRUNE_ARCHIVE (C, Xc, arch_cap)
// C: fitness values of non-dominated solutions; Xc: non-dominated solutions; arch_cap: maximum
capacity of the archive; B: the number of the non-dominated solutions //
CDA = zeros(B)              //CDA: crowding distance; initialize as a 2D matrix//
FOR k = 1 TO K              // K: number of objectives//
      C_k = C(k)            //consider the fitness value for the kth objective//
      [C_k_sort, sorted_indices] = sort(C_k)  //sort the kth objective in ascending order and get
                                                the sorted particle indices//
      CDA(sorted_indices_first,k) = 10000      // particle corresponding to the largest objective
                                                function is given a large crowding distance
      CDA(sorted_indices_final,k) = 10000      // particle corresponding to the smallest objective
                                                function is also given a large crowding distance
       FOR b = 2 to (B-1)      // the 1st and the last ones are excluded//
            CDA(sorted_indices(b)) = CDA(sorted_indices(b))
                                   +(C_k_sort(b+1) – C_k_sort(b-1))/(C_k_sort(1) –
```

```
                                             C_k_sort(end))
                         //crowding distance calculation - normalized//
    END FOR
END FOR
[CDA, particle_indices_sorted] = sort (CDA)     // Sort in descending order using each objective
                                                   value//
particle_indices_pruned = particle_indices_sorted(1: arch_cap)  // Retain the first (number of
                                                   solutions = maximum capacity
                                                   of archive) with the largest
                                                   crowding distance values in the
                                                   archive//
Out ← particle_indices_pruned      // output the Pareto (non-dominated) optimal solutions //
```

Compromise Solution

To avoid the subjective judgment of decision makers, a fuzzy set function is employed to mimic the agency preferences and to find the compromise solution from the non-dominated solutions in the archive. Therefore, at the final generation of algorithm, the compromise solution is identified from the equation (17) (*43*):

$$\mu_k^i = \begin{cases} 1, & F_k(X_i) \le F_k^{min} \\ \frac{F_k^{max} - F_k(X_i)}{F_k^{max} - F_k^{min}}, & F_k^{min} < F_k(X_i) < F_k^{max} \\ 0, & F_k(X_i) \ge F_k^{max} \end{cases} \quad (17)$$

Where μ_k^i = membership value of the *k*th objective function and particle *i*th, X_i = non-dominated solution ith in the archive, $F_k^{min}$and F_k^{max} = the minimum and maximum of the kth objective function.

Then, the normalized fuzzy set function μ^i of non-dominated solution ith is estimated by:

$$\mu^i = \frac{\sum_{k=1}^{K} \mu_k^i}{\sum_i^B \sum_{k=1}^{K} \mu_k^i} \quad (18)$$

Where *K* = the total number of objectives, *B* = the total number of the non-dominated solutions in the archive.

The particle ith having the maximum μ^i in the archive is selected as the compromise solution (*43*).

IMPLEMENTATION OF THE PROBLEM

The developed DBB-MOPSO algorithm is applied to a pavement maintenance decision optimization problem. This problem is the selection of the optimal treatment action from 5 maintenance actions for 5 pavement sections over 10 years. The decision variables are encoded by direct and indirect representations as shown in Figure 4.

There are Several variations of PSO used to solve continuous problems. The binary version of PSO was developed by Kennedy and Eberhart. The discrete multi-objective particle swarm (DMOPSO) algorithm presented by Izakian et al. (2010) is the same original version for multi objective problem. DMOPSO is used to evaluate the performance of the developed algorithm. Both the proposed algorithm and the one Izakian et al. are coded in MATLAB and applied to the same optimization problem. The parameters of the problem are given below:

- A swarm size of 100, archive size of 100, number of iterations of 100 are assumed for both algorithms.
- a velocity range [6, -6], is assumed for the DMOPSO algorithm. This velocity range is recommended by Kennedy and Eberhart (1995) for discrete problems. The values of c_1 = 2, c_2 = 2 are recommended by Izakian *et al*. (2010).

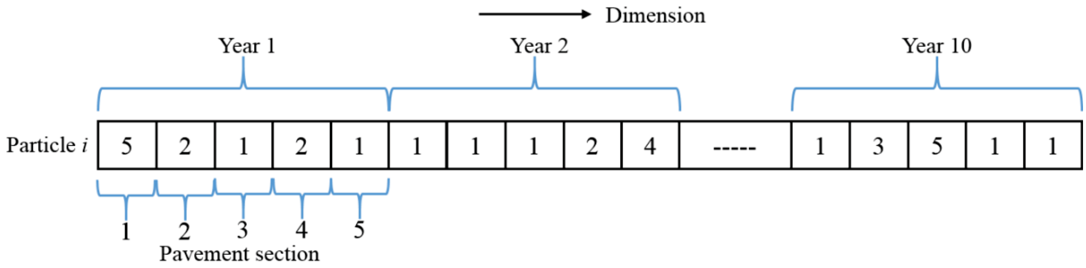

FIGURE 4 Particle position encoding for the pavement maintenance optimization problem

PERFORMANCE METRICS

There are different metrics to examine the accuracy and the diversity of different procedures in regenerating the Pareto front of multi-objective optimization problems. Some of these metrics are described below, before employing these to perform an evaluation of the effectiveness of the proposed method.

Maximum Spread

This measure was developed by Zitzler et al. (2000). "This index is utilized to estimate the maximum extension covered by the non-dominated solutions in the Pareto front. In a two objective problem, the Maximum Spread corresponds to the Euclidean distance between the two farther solutions" (*47*, *48*).

$$MS = \sqrt{\sum_{k=1}^{K}\left[\max(f_k^b) - \min(f_k^b)\right]^2} \qquad \forall b \in \{1, 2, \dots, B\} \qquad (17)$$

where B = the number of the non-dominated solutions, K = the total number of objectives. Larger values of this index indicate better performance.

Spacing

Spacing is a measure to determine how well distributed (spaced) the solutions are in the non-dominated set obtained. It is defined as:

$$S = \sqrt{\frac{1}{D}\sum_{i=1}^{D}(q_i - \bar{q})^2} \qquad (18)$$

where qi = the minimum value of the sum of the absolute difference for every objective function value between the ith solution and all the D non-dominated solutions found. In Equation (18),

$$q_i = min_{l=1 \wedge l \neq i}^{D}\left(\sum_{k=1}^{K}\left|f_k^i - f_k^l\right|\right) \qquad (19)$$

The $\bar{q}$ = mean of all q_i, and is defined as:

$$\bar{q} = \sum_{i=1}^{D} {q_i}/{D} \qquad (20)$$

If the value of this metric is smaller, the solutions will be uniformly spaced (*48*).

Generational Distance (GD)

Generational distance was proposed by Van Veldhuizen and Lamont (1998). It is a method to evaluate the Euclidean distance between each element in the non-dominated solution found until now and its nearest element in the Pareto-optimal set. It is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{D} d_i^2}}{D} \qquad (23)$$

where D = the number of members in the set of non-dominated solutions found to date, d_i = the Euclidean distance between non-dominated solutions (measured in the objective function space).

All members found are in the Pareto-optimal set if the GD value is equal to zero (*49*).

Diversity (D)

The diversity metric was developed by Deb et al. (2002). It is used to estimate the extent of spread among the found solutions. It is defined as follows (*49*):

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{D-1} |d_i - \bar{d}|}{d_f + d_l + (D-1)\,\bar{d}} \tag{24}$$

$$\text{Where } \bar{d} = \frac{\sum_{i=1}^{D-1} d_i}{D-1} \tag{25}$$

d_f, d_l are the Euclidean distances between the extreme solutions and the boundary, non-dominated solutions (first and final solutions of the found non-dominated set), $\bar{d}$ = the average of all distances di, i = 1, 2, ..., (*D* - 1), assuming that there are *D* solutions on the best non-dominated front.

RESULTS

The discrete barebones multi objective particle swarm optimization (DBB-MOPSO) is applied to find the optimal maintenance action plan for five pavement sections over 10 years. For algorithm implementation, the program code in MATLAB is generated. After 100 generations, for the DBB-MOPSO algorithm, 10 non-dominated solutions from 100 solutions are found as shown in Figure 5. The efficiency of the proposed algorithm is evaluated, as mentioned earlier, by comparing it against the existing algorithm called the discrete multi-objective particle swarm (DMPOSO) algorithm developed by Izakian et al. (2010). After 100 generations, for the DMOPSO algorithm, 17 non-dominated solutions were from 100 solutions found as shown in Figure 6. The DMOPSO needs the lowest execution time about 27 hours to achieve results compared to the novel algorithm about 34.5 hours but the novel algorithm converges to optimal solutions with lower generations.

To simulate the agency preferences, the compromise solution is applied for both algorithms as shown in Figures 5 and 6. The solution having the maximum membership value (μ^i) in the archive is selected as the optimal pavement maintenance in both algorithms. Table 2 shows the optimal maintenance programming found by both the algorithms. It can be seen that the overall value of pavement conditions found by the DBB-MOPSO algorithm is slightly better than the overall value of pavement conditions found by DMOPSO, but the cost value of DMOPSO is about 5% better than the proposed algorithm. In the optimal maintenance plan found by DBB-MOPSO algorithm as shown in Table 3, there is heavier investment in the pavement maintenance of all sections at the beginning of the plan period compared with the end of the 10 years. However, in optimal maintenance program found by DMOPSO algorithm as shown in Table 4, there is heavy maintenance investment for most sections in the middle years.

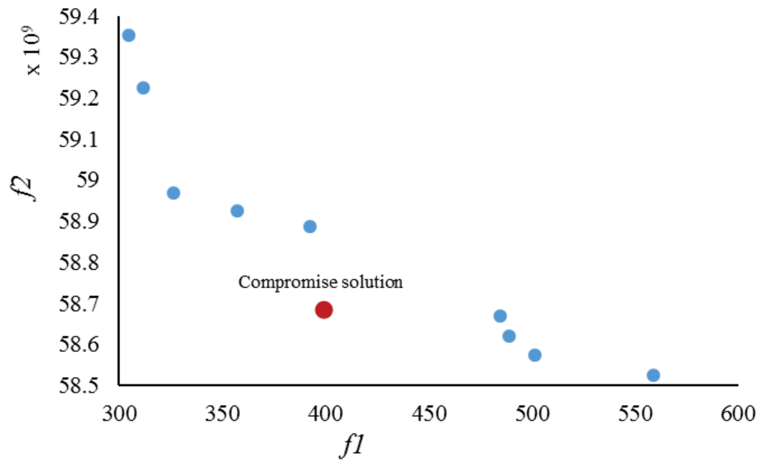

FIGURE 5 Pareto solutions of the DBB-MPSO at 100 generations

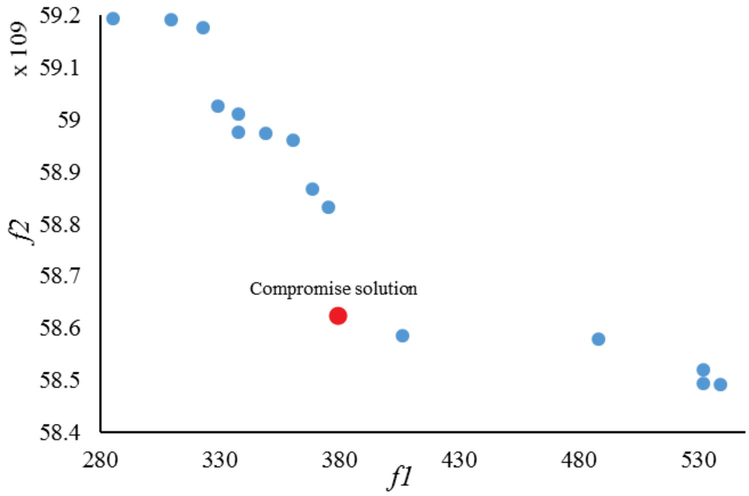

FIGURE 6 Non-dominated solutions of the DMOPSO at 100 generations

TABLE 2 Optimal Maintenance Plans Found by Both Algorithms

Algorithm	Cost	Condition	μ
DBB-MOPSO	399.25	5.87E+10	0.121
DMOPSO	379.22	5.86E+10	0.077

TABLE 3 The Pavement Maintenance Programming Based on The DBB-MOPSO Algorithm

M&R action	Year 1					Year 2					Year 3					Year 4					Year 5				
	Sections																								
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
3	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0
4	0	0	1	0	0	1	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0
5	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	1	1

M&R action	Year 6					Year 7					Year 8					Year 9					Year 10				
	Sections																								
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	1	0	0	0
2	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
3	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
4	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1
5	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE 4 The Pavement Maintenance Programming Based on The DMOPSO Algorithm

M&R action	Year 1					Year 2					Year 3					Year 4					Year 5				
	Sections																								
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
4	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
5	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	1

M&R action	Year 6					Year 7					Year 8					Year 9					Year 10				
	Sections																								
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0
2	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	1
3	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
4	1	0	0	0	0	0	0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0
5	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

To exam the performance of the novel algorithm, the performance metrics with respect to spacing, maximum spread, generational distance (GD), and diversity are estimated. Table 5 shows the results reported in terms of the mean and standard deviation of the performance metrics for both algorithms. There is no significant difference in the mean and variance for the DBB-MOPSO and the DMOPSO algorithms at 100 iterations.

TABLE 5 The Mean and Variance of Different Performance Metrics Over 100 Iterations

Algorithms		Performance Metrics			
		Spacing	Maximum Spread	Generational distance	Diversity
DBB-MOPSO	Mean	2.18E+08	7.85E+08	1.07E+07	0.971
	SD	4.86E+07	1.87E+08	1.01E+07	0.0131
DMOPSO	Mean	2.31E+08	8.41E+08	1.02E+07	0.973
	SD	3.97E+07	1.33E+08	6.32E+06	0.0112

For verifying the non-dominated solutions spread in the entire region of the true front, the diversity measure is estimated. Figure 7 shows the proposed algorithm has lower diversity at 100 generations compared to the DMOPSO algorithm hence the latter performs better. However, when the mean value of diversity over the 100 iterations is considered, as shown in Table 5, the difference is very small. The larger standard deviation value for the proposed algorithm means that the diversity of solutions is not as steady as for DMOPSO. As shown in Figure 8, DBB-MOPSO has slightly smaller value of generational distance GD compared to DMOPSO at the 100th iteration. Therefore, the convergence speed of the DBB-MOPSO to the Pareto front is slightly better than the DMOPSO at this stage. But the average GD over the iterations is very similar (Table 5). According to Figure 9, the maximum spread of the DBB-MOPSO algorithm is approximately in the same range of DMOPSO, but the mean value of this performance metric over the whole iteration range is definitely smaller than that of DMOPSO. Figure 10 shows that DBB-MOPSO has slightly smaller values of spacing. The smaller values means the solutions of DBB-MOPSO are more uniformly spaced compared to the DMOPSO algorithm and this is an advantage of the proposed algorithm.

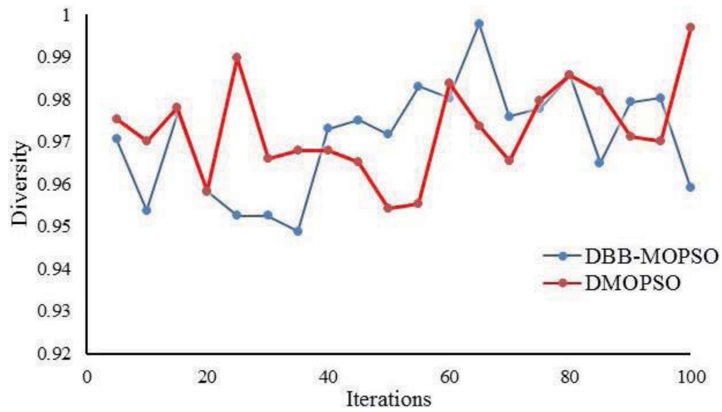

FIGURE 7 The diversity metric of the both algorithms

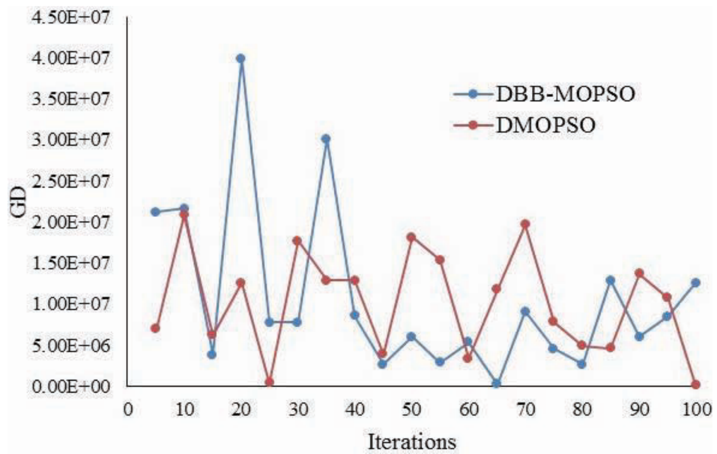

FIGURE 8 The generational distance metric of the both algorithms

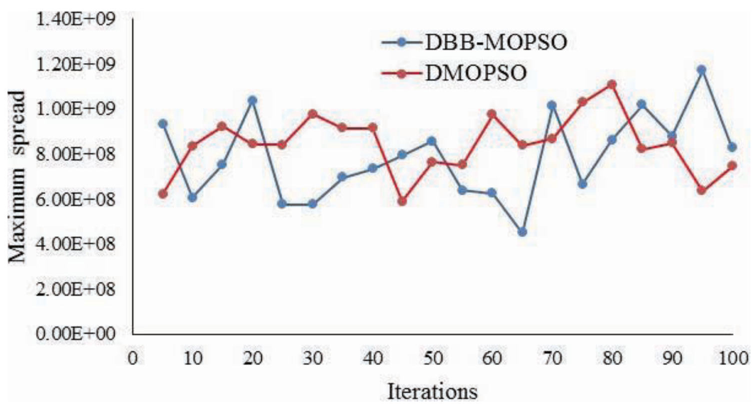

FIGURE 9 The maximum spread metric of the both algorithms

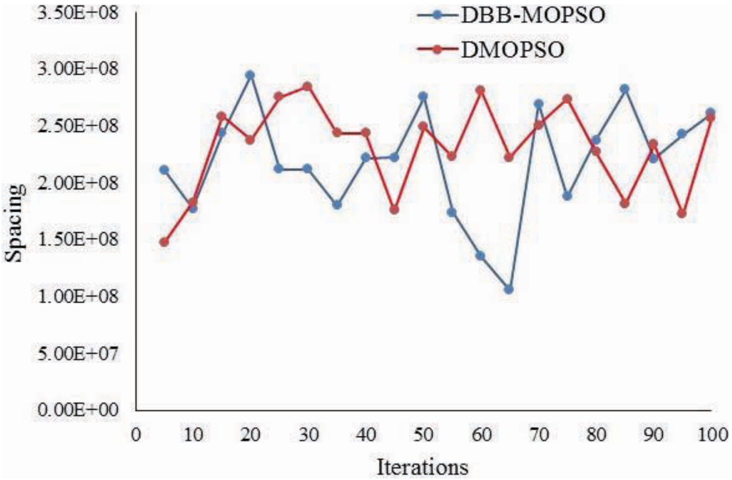

FIGURE 10 The spacing metric of the both algorithms

CONCLUSION

A novel particle swarm algorithm is developed for a discrete multi-objective problem. This novel algorithm, being based on the bare-bones method, is parameter free presenting a clear advantage over the algorithms where the user has to do parameter selection. The proposed algorithm is applied to find optimal rehabilitation scheduling considering the two objectives the minimization of the total pavement rehabilitation cost and the minimization of the sum of all residual PCI values.

Although the results showed that the cost obtained via the proposed algorithm is slightly higher than that of the DMOPSO algorithm, the overall value of pavement performance found by DBB-MOPSO is higher than that obtained by DMOPSO, another existing discrete optimization algorithm. The optimal maintenance plan found by the DMOPSO algorithm is comparatively similar to that found by DBB-MOPSO, but the results showed that the novel algorithm can converge to Pareto front with little iterations, lower diversity, smaller GD, and higher maximum spread compared to the DMOPSO algorithm.

In future, the novel algorithm will be put through more validation by benchmarking its performance with different algorithms from the particle swarm optimization and genetic algorithm domains. Moreover, in this paper, the algorithm is applied to an unconstrained pavement maintenance decision optimization problem. In the future, it will also be tested on a constrained problem of pavement maintenance programming. This algorithm was applied to a small test case for validation. Large networks will be tested in future.

ACKNOWLEDGEMENT

The authors wish to thank the Ministry of Higher Education and Scientific Research, Baghdad, Republic of Iraq for support this research.

REFERENCES

1. Chen, C., G. W. Flintsch, and I. L. Al-Qadi. 6th International Conference on Managing Pavements, 2004.
2. Alsherri, A., and K. P. George. Reliability Model for Pavement Performance. J. Transp. Eng., Vol. 114, No. 2, 1988, pp. 294–306.
3. Scott Wilson Pavement Engineering Ltd. Research on Using TTS Data for Maintenance Management of Local Roads Deterioration Models, Whole Life Cost and Economic Projection Models. 2005.
4. Haas, R., W. R. Hudson, and J. P. Zaniewski. Modern Pavement Management. Krieger Pub. Co., Cornell University, . 2nd Ed. 1994.
5. Meneses, S. C. N., A. J. L. Ferreira, and A. C. Collop. Multi-Objective Decision-Aid Tool for Pavement Management. ICE - Transp., Vol. 166, No. 2, 2012, pp. 1–16.
6. Fwa, T. F., W. F. Chan, and K. Z. Hoque. Multi-Objective Optimisation for Pavement Maintenance Programming. J. Transp. Eng., Vol. 126, No. 5, 2000, pp. 367–374.

7. Elhadidy, A. A., E. E. Elbeltagi, and M. A. Ammar. Optimum Aanalysis of Pavement Maintenance Using Multi-Objective Genetic Algorithms. HBRC J., 2014, .
8. Chan, W. T., T. F. Fwa, and C. Y. Tan. Road Maintenance Planning Using Genetic Algorithms I: Formulation. J. Transp. Eng., Vol. 120, No. 5, 1994, pp. 693–709.
9. Fwa, T. F., C. Y. Tan, and W. T. Chan. Road Maintenance Planning Using Genetic Algorithm. J. Transp. Eng., Vol. 120, No. 5, 1994, pp. 710–722.
10. Fwa, T. F., W. T. Chan, and C. Y. Tan. Genetic-Algorithm Programing of Road Maintenance and Rehabilitation. J. Transp. Eng., Vol. 122, No. 3, 1996, pp. 246–253.
11. Fwa, T. F., W. T. Chan, and K. Z. Hoque. Analysis of Pavement Management Activities Programming by Genetic Algorithms. Transp. Res. Rec. J. Transp. Res. Board, Vol. 1643, No. 1, 1998, pp. 1–6.
12. Chan, W. T., T. . Fwa, and K. Zahidul Hoque. Constraint Handling Methods in Pavement Maintenance Programming. Transp. Res. Part C Emerg. Technol., Vol. 9, No. 3, 2001, pp. 175–190.
13. Ferreira, A., A. Antunes, and L. Picado-Santos. Probabilistic Segment-Linked Pavement Management Optimisation Model. J. Transp. Eng., Vol. 128, No. 6, 2002, pp. 568–577.
14. Chan, W. T., T. F. Fwa, and J. Y. Tan. Optimal Fund-Allocation Analysis for Multidistrict Highway Agencies. J. Infrastruct. Syst., Vol. 9, No. 4, 2003, pp. 167–175.
15. Cheu, R. L., Y. Wang, and T. F. Fwa. Genetic Algorithm-Simulation Methodology for Pavement Maintenance Scheduling. Comput. Civ. Infrastruct. Eng., Vol. 19, No. 6, 2004, pp. 446–455.
16. Morcous, G., and Z. Lounis. Maintenance Optimisation of Infrastructure Networks Using Genetic Algorithms. Autom. Constr., Vol. 14, No. 1, 2005, pp. 129–142.
17. Chootinan, P., A. Chen, M. R. Horrocks, and D. Bolling. A Multi-Year Pavement Maintenance Program Using A Stochastic Simulation-Based Genetic Algorithm Approach. Transp. Res. Part A Policy Pract., Vol. 40, No. 9, 2006, pp. 725–743.
18. Jha, M. K., and J. Abdullah. A Markovian Approach for Optimizing Highway Life-Cycle with Genetic Algorithms by Considering Maintenance of Roadside Appurtenances. J. Franklin Inst., Vol. 343, No. 4-5, 2006, pp. 404–419.
19. Farhan, J., and T. F. Fwa. Incorporating Priority Preferences into Pavement Maintenance Programming. J. Transp. Eng., Vol. 138, No. 6, 2012, pp. 714–722.
20. Jawad, D., and K. Ozbay. 85th Annual Meeting of the Transportation Research Board, 2006.
21. Pilson, C., W. R. Hudson, and V. Anderson. Multiobjective Optimisation in Pavement Management by Using Genetic Algorithms and Efficient Surfaces. Transp. Res. Rec. J. Transp. Res. Board, Vol. 1655, No. 99, 1999, pp. 42–48.
22. Herabat, P., and A. Tangphaisankun. Multi-Objective Optimisation Model using Constraint-Based Genetic Algorithms for Thailand Pavement Management. J. East. Asia Soc. Transp. Stud., Vol. 6, 2005, pp. 1137–1152.
23. Golroo, A., and S. L. Tighe. Optimum Genetic Algorithm Structure Selection in Pavement Management. Asian J. Appl. Sci. 5, Vol. 5, No. 6, 2012, pp. 327–341.
24. Chikezie, C. U., A. T. Olowosulu, and O. S. Abejide. Multiobjective Optimisation for Pavement Maintenance and Rehabilitation Programming Using Genetic Algorithms. Arch. Appl. Sci. Res., Vol. 5, No. 4, 2013, pp. 76–83.
25. Wang, Y., and S. Goldschmidt. 30th Conference of Australian Institutes of Transport Research, 2008.
26. Shen, Y., Y. Bu, and M. Yuan. 4th IEEE Conference on Industrial Electronics and Applications, 2009.

27. Tayebi, N. R., F. Moghadasnejhad, and A. Hassani. International Conference on Advances in Electrical & Electronics, 2010.
28. Chou, J.-S., and T.-S. Le. Reliability-Based Performance Simulation for Optimized Pavement Maintenance. Reliab. Eng. Syst. Saf., Vol. 96, No. 10, 2011, pp. 1402–1410.
29. Chen, C. Soft Computing-based Life-Cycle Cost Analysis Tools for Transportation Infrastructure Management. Virginia Polytechnic Institute and State University, 2007.
30. FHWA. Life-Cycle Cost Analysis in Pavement Design. Washington, DC, 1998.
31. Javed, F. Integrated Prioritisation and Optimisation Approach for Pavement Management. National University of Singapore, 2011.
32. Shahin, M. Y. Pavement Management for Airports, Roads and Parking Lots.. Springer, New York, NY, USA, . 2nd Ed. 2005.
33. Mahmood, M. the Inaugural College of Art, Design & Built Environment Doctoral Conference, 2014.
34. Kennedy, J., and R. Eberhart. International Conference on Neural Networks, 1995.
35. Rao, S. S. Engineering Optimisation: Theory and Practice. John Wiley & Sons, Inc., Hoboken, New Jersey, . 4th Ed. 2009.
36. Teodorović, D. Swarm Intelligence Systems for Transportation Engineering: Principles and Applications. Transp. Res. Part C Emerg. Technol., Vol. 16, 2008, pp. 651–667.
37. De Carvalho, A. B., and A. Pozo. Measuring the Convergence and Diversity of CDAS Multi-Objective Particle Swarm Optimisation Algorithms: A Study of Many-Objective Problems. Neurocomputing, Vol. 75, No. 1, 2012, pp. 43–51.
38. Osyczka, A. An Approach to Multicriterion Optimisation Problems for Engineering Design. Comput. Methods Appl. Mech. Eng., Vol. 15, 1978, pp. 309–333.
39. Engelbrecht, A. P. Computational Intelligence: An Introduction. John Wiley & Sons, . 2nd Ed. 2007.
40. Liao, C.-J., C.-T. Tseng, and P. Luarn. A Discrete Version of Particle Swarm Optimisation for Flowshop Scheduling Problems. Comput. Oper. Res., Vol. 34, No. 10, 2007, pp. 3099–3111.
41. Pugh, J., and A. Martinoli. Swarm Intelligence Symposium, 2006.
42. Kennedy, J., and R. C. Eberhart. International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, 1997.
43. Zhang, Y., D.-W. Gong, and Z. Ding. A Bare-Bones Multi-Objective Particle Swarm Optimisation Algorithm for Environmental/Economic Dispatch. Inf. Sci. (Ny)., Vol. 192, 2012, pp. 213–227.
44. Izakian, H., B. T. Ladani, and A. Abraham. A Discrete Particle Swarm Optimisation Approach for Grid Job Scheduling. Int. J. Innov. Comput. Inf. Control, Vol. 6, No. 9, 2010, pp. 1–9.
45. Mostaghim, S., and J. Teich. Swarm Intelligence Symposium, 2003.
46. Silva, D. R. C., and C. J. A. Bastos-filho. The Second International Conference on Intelligent Systems and Applications, 2013.
47. Santana, R. a., M. R. Pontes, and C. J. a. Bastos-Filho. Ninth International Conference on Intelligent Systems Design and Applications, 2009.
48. Salazar-Lechuga, M., and J. E. Rowe. Congress on Evolutionary Computation, 2005.
49. Tsai, S.-J. et al. An Improved Multi-Objective Particle Swarm Optimizer for Multi-Objective Problems. Expert Syst. Appl., Vol. 37, No. 8, 2010, pp. 5872–5886.