A Parameter-Free Discrete Particle Swarm Algorithm and Its Application to Multi-Objective Pavement Maintenance Schemes

Maher Mahmood[a,*], Senthan Mathavan[b], Mujib Rahman[c]

[a] *Civil Engineering Department, Faculty of Engineering, University of Anbar, Ramadi, Iraq*

[b] *School of Architecture, Design and the Built Environment, Nottingham Trent University, Nottingham, United Kingdom*

[c] *Senior Lecturer, Department of Civil Engineering, Brunel University, Uxbridge, United Kingdom*

Abstract: Regular maintenance is paramount for a healthy road network, the arteries of any economy. As the resources for maintenance are limited, optimization is necessary. A number of conflicting objectives exist with many influencing variables. Although many methods have been proposed, the related research is very active, due to difficulties in adoption to the actual practice owing to reasons such high-dimensional problems even for small road networks. Literature survey tells that particle swarms have not been exploited much, mainly due to unavailability of many techniques in this domain for multi-objective discrete problems like this. In this work, a novel particle swarm algorithm is proposed for a general, discrete, multi-objective problem. In contrast to the standard particle swarm, the bare-bones technique has a clear advantage in that it is a parameter-free technique, hence the end users need not be optimization experts. However, the existing barebones algorithm is available only for continuous domains, sans any particle velocity terms. For discrete domains, the proposed method introduces a parameter-free velocity term to the standard bare-bones algorithm. Based on the peak velocities observed by the different dimensions of a particle, its new position is calculated. A number of benchmark test functions are also solved. The results show that the proposed algorithm is highly competitive and able to obtain much better spread of solutions compared to three other existing PSO and genetic algorithms. The method is benchmarked against a number of other algorithms on an actual pavement maintenance problem. When compared against another particle swarm algorithm, it not only shows better performance, but also significant reduction in run-time compared to other POS algorithm. Hence, for large road network maintenance, the proposed method shows a lot of promise in terms of analysis time, while improving on the quality of solutions.

Keywords: multi-objective optimization; discrete optimization; particle swarm optimization; bare-bones; pavement maintenance; pavement management.

*Corresponding Author
E-mail address: maher.mahmood@uoanbar.edu.iq.

1 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 ever increasing resource deficiency, maintenance activity 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 essential tools that ensure that all pavement sections are maintained at adequately high service levels with a low budget and resources usage, without causing any significant negative effect on environment, safe traffic operations and social activities. At both network level and project level, many highway agencies employ prioritization programming models to compare pavement investment alternatives. 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 conducted based on this priority-ranking index [3].

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 [4]. Optimization has been widely used in research 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 experimented with [4].

A variety of stochastic and deterministic programming techniques have been developed for finding optimal pavement maintenance plan. Stochastic programming techniques such as Markov decision process were used to overcome a deficiency of data availability [5]. To address particular optimisation problems, deterministic programming techniques such as linear or nonlinear programming [6] quadratic programming [7], integer programming [8], dynamic programming [9] and robust optimisation [10] were used.

Many researchers use genetic algorithm for single and multi-objective optimization for pavement decision making problems: single objective GA [11], and Multi objective GA [12]. When it comes to using PSO for pavement problems, Wang and Goldschmidt [13] 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. Cluster models with similarity and dissimilarity analysis were employed in the project interaction pre-optimization process to avoid roadwork on two paths between origin and destination at similar times. The pre-optimization model was used as an input of a global multi-objective optimization model-based particle swarm optimization (PSO). The multi-objective PSO problem was converted into a single-objective problem by using the weighted aggregation method [13]. Shen et al. (2009) used chaos particle swarm optimization (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 [14]. In 2010, Tayebi and Hassani used PSO with single-objective function scenarios for a pavement management system at the network level [15]. The same hypothetical problem formulation of the Pavenet_R model by [16] was used to apply a PSO algorithm for pavement maintenance programming [15]. Chou and Le formulated a multi-objective PSO algorithm (i.e. classical one) to study the effect of overlay maintenance activities on the performance of 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 [17]. Moreira et al. applied two optimization algorithms for pavement maintenance scheduling. The first one was NSGA-II for solving a multi-objective problem, while the second algorithm was a genetic algorithm to optimize a single-objective problem [18]. Santos et al. developed a novel adaptive hybrid genetic algorithm (AHGA) by combining GA with local search (LS) methods for finding the optimal pavement M&R strategy [19]. GA, NSGA-II and single-objective PSO are the state of the art methods for solving pavement maintenance scheduling problem. GA and NSGA-II were applied on pavement maintenance scheduling problem with different problem formulation, different objectives, single or multi objective, or different methods of constraints handling. In addition, PSO was used for solving this problem considering single-objective function. Therefore, it is required to use an optimization algorithm that has few parameters to modify that is easy to implement. In addition, this algorithm has capability to discrete multi-objective optimization problem for pavement maintenance management.

Despite the research efforts with numerous methods, their adoption in engineering practice has been difficult. The main reason for the lack of application has been due to the long times taken by the algorithms for the typically large dimensions encountered in pavement scheduling, i.e. many pavement sections and the associated treatment decision variables covering multiple time periods. In addition, many more objectives, such as impacts on society, the environment and mobility are given consideration in the modern era, requiring a truly multi-objective approach for maintenance scheduling [20]. Hence, the speed and performance are paramount for this application area and the drive to achieve these keeps the research domain very active to this day. As seen earlier, many evolutionary approaches, such as genetic algorithms (GAs) and particle swarm optimizers (PSOs), have been experimented to overcome the aforesaid issues. GAs, especially, have been extensively researched about for pavement maintenance and significant improvements have in terms of speed. However, GAs require the selection of good numerical values for certain parameters, such as mutation and crossover operators, making their use both subjective and, for non-experts, non-optimal. In comparison to GA, PSO has not been studied as much as shown above. In addition, regular PSO algorithms are also plagued by their dependence on numerical values chosen for certain parameters, e.g. the acceleration coefficients of the particles.

This is the context in which another PSO method, called bare-bones, is proposed for PMS. Bare-bones PSO [21] unlike other PSO methods, is a parameter-free technique where the user does not have to choose values for parameters involved. However, the proposed method [21] is for continuous search domains, whereas pavement maintenance consists of various decisions that are essentially discrete in nature. For example, there could be two decisions like ‘no actions needed’ and ‘repave with a 5 mm thick layer’, and these are essentially linguistic quantities that, with some effort, can be converted into discrete values. Hence, the proposed method should be usable with discrete search spaces. This paper proposes an alteration to the original proposal whereby discretization is possible. In addition, the existing barebones PSO algorithm, in contrast to the regular PSO, does not use any velocity terms for the particles. This paper shows why velocity terms are needed for discrete domains, and also proposes and implements a way to reintroduce them. In this regard, the new algorithm exploits the strengths of the regular and barebones methods to arrive at a feasible solution method, bringing forth a significant novelty.

The paper continues as follows. The next section introduces review on MOEA for discrete Problems. Then a section introduces the basics of the maintenance optimization.Thereafter, a section is dedicated to explaining the essentials of PSO. Section 4 provides the theory of the proposed algorithm. Then a section briefly outlines some performance metrics that are used to evaluate the solutions obtained via different methods. This is followed by a part containing the numerical details of the actual problem tackled. An extensive section is devoted to analysing the results obtained via the proposed algorithm and puts in the perspective of those obtained via three other methods followed by conclusions.

2 STATE OF THE ART MOEA METHODS FOR DISCRETE OPTIMIZATION PROBLEMS

There are number of evolutionary algorithms that have been widely used to solve continuous and discrete optimization problems. The most popular MOEA that has been used and applied successfully to solve a numerous optimization problems is non-dominated sorting genetic algorithm II (NSGA-II) [22]. For handling many-objective optimization problems, new version of evolutionary optimization algorithm called NSGA-III was proposed by extending the NSGA-II framework [23]. Multi-objective evolutionary algorithm based on decomposition MOEA\D was developed by Zhang and Li in 2007 [24]. MOEA\D was improved and applied to solve a multi-objective optimization problem by decomposing it into several single optimization sub-problems by using common aggregation methods [25–32]. To use advantages of various optimization algorithms for dealing with complicated multi-objective optimization problems, numbers of hybrid multi-objective evolutionary algorithms were developed [33–35].

A new heuristic algorithm called multi-objective quantum evolutionary algorithm (MOQEA) was proposed. This algorithm was developed based on the theory of quantum computing that has little q-bit individuals are evolved to find a satisfactory outcome [36]. A novel hybrid discrete differential evolution (HDDE) algorithm was formulated to solve flow shop scheduling problems with makespan criterion. An effective insert neighbourhood based local search algorithm structure was embedded to balance the global exploration and local exploitation [37]. An ensemble of discrete differential evolution DDE algorithms was proposed to solve the generalized traveling salesman problem and evaluated on a set of benchmark problems. The suggested method allow the DDE algorithm to make use crossover operators and number of parameter values simultaneously [38]. A discrete harmony search (DHS) algorithm was developed for solving flexible job-shop scheduling problem (FJSP) to satisfy weighted set of multiple objectives. To improve the local exploitation ability, number of local search methods were included [39].

In addition, a new version of PSO called modified particle swarm optimization for discrete problem suggested. This version is compared to the original version of PSO by splitting the cognitive component of the original version into two different components which can be called good experience component and bad experience component [40]. A PSO algorithm with an event-based heuristic were combined to develop a new improved algorithm called a hybrid particle swarm optimization (HPSO). It was implemented to solve the dynamic and discrete berth allocation problem and the dynamic quay crane assignment problem [41].

Moreover, the proposed a discrete artificial bee colony (DABC) algorithm that were hybridized with an iterated greedy (IG) and iterated local search (ILS) algorithms IG algorithm embedded in the VNS procedure based on swap and insertion neighbourhood structures. The key aim of the hybridization is for performing the global search by the exploitation of the search space and increasing the search on the local minima and then achieving the balance in both global and local search effectively [42]. A discrete artificial bee colony (DABC) algorithm was proposed to solve the lot-streaming flow shop scheduling problem to satisfy the objective that is minimization of total weighted earliness and tardiness penalties [43]. A new version of the artificial fish swarm algorithm was developed to solve binary optimization problem. The proposed algorithm was applied on multidimensional knapsack problem (MKP) and then tested on a set of benchmark functions [44]. The chaos driven discrete artificial bee colony (CDABC) algorithm was developed. Four unique chaos maps of Burgers, Lozi, Delayed Logistic and Tinkerbell were included to measure the effectiveness of applying these instead of Mersenne Twister as chaos pseudo-random number generators [45]. A nature-inspired algorithm called honeybee mating optimization HBMO algorithm was developed. It was applied to address a related routing issue to the personal rapid transit (PRT) system [46]. Another swarm intelligence (SI) algorithm called dragonfly algorithm (DA) was proposed to solve binary and multi-objective optimization problems. The main characteristics of dragonfly algorithm were found to explore and exploit the search space respectively [47].

Furthermore, a Binary Grey Wolf Optimizer application (BGWO) was presented to find the commitment program of unit commitment UC problem. It was applied to satisfy just a single objective which is a cost minimization associated with bound, equality and inequality constraints [48]. A new version of Ant Lion Optimizer called Multi-Objective Ant Lion Optimizer (MOALO) was proposed. To test the effectiveness of the developed algorithm, is employed. Also, the algorithm was applied to a number of multi-objective engineering design problems and also a set of unconstrained and constrained benchmark functions [49]. A new algorithm called multi-objective discrete virus optimization algorithm (MODVOA) was developed for solving the flexible job-shop scheduling problem with controllable processing times (MOFJSP-CPT). It was applied to satisfy the objectives which are minimization of the makespan and the total additional resource consumption [50].

3 DESCRIPTION OF THE PAVEMENT MAINTENANCE DECISION PROBLEM

3.1 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 the same length [3,16]. The segments are assumed to have uniform conditions throughout their span, making it suitable for the M&R scheme to prescribe a single maintenance decision for that segment.

The agency costs of highway assets consist of the expenses for maintenance, rehabilitation and reconstruction. Rehabilitation is necessary for the highway asset at least once over its lifetime to keep it above the minimum acceptable serviceability and safety levels. The cost of any particular rehabilitation activity, usually a form of construction, comes from: materials, preliminary engineering, and construction management [51]. 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:

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

where PWF is the present worth factor, given by:

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

The typical range of discount rates R recommended by FHWA is 3% to 5% [52], t = time at which the money is spent (specified in years). In this work, a value of 4% is used for R.

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 warning level 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 unit study period, which might be a week, a month, or a year, is selected depending on the requirements of the highway authority [16].

3.2 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 to form a single objective by allocating proper weighting factors to each [16].

The main challenge in pavement management is the selection of maintenance investment alternatives for a large number of pavement sections over multiple time periods [53]. 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. Therefore, a multi-objective programming of pavement management activities is required. It can be presented mathematically as the following:

The first objective is to minimize the total pavement maintenance cost.

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

The second objective is to minimize the sum of all residual pavement condition index (PCI) values.

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

where,

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

In the equations above, *m* is the treatment type; *M* stands for the total number of available treatment options; *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*. For the problem treated in this paper, L_p = 152.5, W_p = 3.6 and C_m = [0, 10, 33, 41, 78], the last being the cost values for the 5 treatment actions given in Table 1.

In this work, the following acceptable level for section performance is chosen: $\text{PCI}_{p,t} \geq 65$ %.

Table 1: Rehabilitation options in consideration

3.3 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 [54]. In general, deterioration models are established in terms of a pavement condition indicator and the exogenous influences contributing to pavement deterioration [55]. Various researchers have developed network-level deterministic deterioration prediction models for flexible pavement, 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 [56] :

$$\mathrm{PCI} = 97.744 - 0.15\,X_5 - 0.064\,X_4 - 0.515\,X_2 + 3.748\,X_3 \quad (5)$$

where X_1 is the cumulative equivalent single axle load (ESAL); X_2 is the pavement age; X_3 is the maintenance effect (inlay and overlay thickness, in inches, estimated with the help of Table 1); X_4 is the total longitudinal and transverse cracking length, in inches; X_5 is the cracking area (alligator, edge, and block), in square inches.

4 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. 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 an operator that is directed by local and global elements. Each solution or particle is assumed to have a position and a velocity. The position and velocity of the i^{th} 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 $n = N\times T$. Then, each particle i updates the position and velocity of its j^{th} dimension at iteration $(z + 1)$ by using the following equations [40,57–59]:

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

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

where, $Pbest_{i,j}(z)$ is the local or personal best position for the j^{th} 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]. As noted earlier, it is up to the user to select the values for w, c_1, and c_2. However, these parameters are strongly coupled to the nature of the problem and the search space, and choosing good values are not very easy. The values chosen for these parameters greatly influence optimization performance [60]. This is a significant drawback, introducing subjectivity, and this underpins the importance for parameter-free methods, such as the one proposed in this paper.

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 [61].

4.1 Multi-Objective Optimization Problems

Multi-objective optimization problems consider the simultaneous satisfaction of two or more objective functions. Furthermore, these multiple objectives 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 [61,62].

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 a least one other objective [63]. The Pareto optimality is defined as, "A decision vector, $\boldsymbol{x}^* \in \mathcal{F}$, is Pareto-optimal if there does not exist a decision vector, $\boldsymbol{x} \neq \boldsymbol{x}^* \in \mathcal{F}$ that dominates it. For maximization problems, this condition can be expressed as, $\nexists k : f_k(\boldsymbol{x}) < f_k(\boldsymbol{x}^*)$. For minimization problems, $\boldsymbol{x}^* \in \mathcal{F}$ will be Pareto-optimal if $f_k(\boldsymbol{x}) > f_k(\boldsymbol{x}^*)$ for any $\boldsymbol{x} \neq \boldsymbol{x}^* \in \mathcal{F}$. An objective vector, $\boldsymbol{f}^*(\boldsymbol{x})$, is Pareto optimal if $\boldsymbol{x}$ is Pareto optimal" [64].

For a set of objective functions $\{f_1, f_2, \ldots, f_K\}$, the condition that a feasible solution $\boldsymbol{x}^*$ dominates another feasible solution $\boldsymbol{x}$, then it is denoted by $\vec{F}(\boldsymbol{x}^*) \prec \vec{F}(\boldsymbol{x})$, the target being maximization.

4.2 Discrete (Binary) Particle Swarm Optimization

Some of 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 decisions will either 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 [65,66].

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 [67]:

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

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

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

4.3 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 BBPSO algorithm is replaced by [21,64]:

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

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, called by BBExp, by modifying the equation thus:

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

As there is a probability of 50% that the j^{th} 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 advantages 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 [21]. In addition, it is easy to implement and performs well when dealing with multi-objective optimization problems [21].

5 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.

Figure 1: Flow chart of the discrete barebones particle swarm optimization algorithm.

5.1 Initialization

5.1.1 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 [21].

One of the main steps in designing an effective particle swarm optimisation 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 [68]. 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 $n = N \times T$. 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 as shown in Figure 2.

Figure 2: Direct representation (encoding) for particle *i*.

In indirect encoding, solutions for each particle are encoded in a position matrix, $n \times M$. 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:

Figure 3: Indirect representation (encoding) for particle *i*.

5.1.2 Particle velocity, local best position

Indirect encoding is used to initialize the velocity of each particle. The $n \times M$ 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 j^{th} dimension of the i^{th} particle and in the swarm. To save the non-dominated solutions found across all iterations, an archive, or memory, is initialized from the initial swarm.

5.2 Updating 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 $X_i(z+1)$ [21]. Usually, one solution is kept as the local best, as the algorithm does not have any means of handling multiple local best solutions. Hence, when our new local position and the current values are non-dominated, we keep the current one intact.

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

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

5.3 Updating 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 optimisation problems the global best position is found in a straightforward manner. Conversely, in multi-objective optimisation problems, the multiple conflicting objectives make it challenging to select a leader solution. To overcome this problem, DBB-MOPSO maintains a memory (archive) with a sufficient capacity to store non-dominated (Pareto) solutions, as proposed by [21,64].

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 $(f_{1,i}, f_{2,i})$, 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,

$$\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2} \qquad (13)$$

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 each particle (a) of the current generation. Then, for each particle in the archive, e, the difference between that and each particle in the current swarm is calculated and the sum of all the squared differences is estimated. The archive particle, g, that has the least sum of all archive particles is chosen as the global best position or the leader particle [69].

5.4 Updating Particle Positions and Velocities

To handle the multi-objective optimisation problem, a new version of BBExp, namely BBVar, has been proposed to update a particle's position by [21], and it works as shown below:

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

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:

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

According to Izakian et al. [2010], the particle's position is proposed to be updated as follows:

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

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 [68]. The same method is used by the DBB-MOPSO algorithm presented here.

5.5 External Archive Pruning

In multi-objective optimisation 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 [21,70]. 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 [21]. The following pseudo-code is the pruning archive procedure.

```
Algorithm: PRUNING ARCHIVE: Out = PRUNE_ARCHIVE
(fitnessNonDominatedSolutions, nonDominatedSolutions, archiveCapacity)
B = number(nonDominatedSolutions)
CDA = zeros(B, totalObjectivesNumber)          //CDA: crowding distance//
FOR k = 1 TO totalObjectivesNumber
        [sortedVal, sortedIndex] = sort(fitnessNonDominatedSolutions (k))
                                                // in  ascending order //
        firstSortedIndex = sortedIndex(1)
        lastSortedIndex = sortedIndex(end)
        CDA(firstSortedIndex, k) = 10000
        CDA(lastSortedIndex, k) = 10000   //the extreme particles are given
                                              a large crowding distance//
         FOR b = 2 to (B-1)
              CDA(sortedIndex (b)) = CDA(sortedIndex (b))
             +(sortedVal (b+1)- sortedVal (b-1))/( sortedVal (1)
                                                    – sortedVal (end)
                                         //crowding distance normalization//
         END FOR
END FOR
[sortedCDA, sortedIndexCDA] = sort (CDA)     // in descending order//
prunedParticleIndices = sortedCDA (1: archiveCapacity)  //Retain the
                                         first archiveCapacity solutions //
Out ← prunedParticleIndices
```

5.6 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 [21]. The procedure of mutation operation is given by the following pseudo-code:

```
Algorithm: MUTATION: mutatedSwarm = MUTATE (currentSwarm,
z)    //z = iteration number//
Z = max. number of iterations
mutatedSwarm  = currentSwarm
FOR i = 1 TO allParticles
      IF $e^{((-8*z)/Z)} > r_4$        // $r_4$ is a quasi-random number in U[0,1] //
            FOR j = 1 TO allDimensions
                  $mutatedSwarm_{i,j}$ == 1       // j is randomly chosen
from {1,2,3,….,M} //
            END FOR
      END IF
End FOR
```

5.7 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 Equation (17) [21]:

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

where, μ_k^i = membership value of the kth objective function and particle *I*; X_i = non-dominated solution *i*th 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 *i* is estimated by:

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

where, K = the total number of objectives; B = the total number of the non-dominated solutions in the archive. The particle having the maximum μ^i, found by Equation (18), in the archive is selected as the compromise solution [21].

5.8 Performance metrics

Quality definition for multi-objective optimization problems is more complex than that for single-objective optimization problems, because the goal of optimization comprises many objectives. There are different metrics to examine the accuracy and the diversity of different procedures in regenerating the Pareto front of multi-objective optimization problems. The following metrics are used in this work to evaluate the quality of the swarm at every iteration. The maximum spread measure was developed by Zitzler et al. [71]. "This index is utilised 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" [72,73]. Spacing is a measure to determine how well distributed (spaced) the solutions are in the non-dominated set obtained. If the value of this metric is smaller, the solutions will be uniformly spaced [73]. Generational Distance (GD) 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. All members found are in the Pareto-optimal set if the GD value is equal to zero [71,74]. The diversity (D) metric was developed by Deb et al. [2002]. It is used to estimate the extent of spread among the found solutions [74,75].

Inverted Generational Distance (IGD) is for evaluating the quality of solutions at any iteration. This metric is given as:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{19}$$

Where n is the number of elements in the true Pareto front, and d_i is the Euclidean distance (measured in objective space) between each of these and the nearest member of the set of non-dominated vectors found by the algorithm. Hence, all the solutions generated are in the true Pareto front of the problem if the value of IGD equal to zero [71,76]. The hypervolume indicator is another quality measure that is widely used. "Given a set of points $X \subset R^d$ and a reference point $r \in R^d$, the hypervolume indicator $H(X)$ is the
Lebesgue measure, $\lambda(.)$, of the region dominated by X and bounded above by r, i.e. $H(X) = \lambda\,(\{q \in R^d \mid \exists\, p \in X: p \le q \wedge q \le r\})$.

Alternatively, the hypervolume indicator may be written as the measure of the union of n isothetic hyperrectangles in d dimensions"[77,78]:

$$H(X) = \lambda\left(\bigcup_{\substack{p\in X \\ p \le r}} [p,r]\right) \tag{20}$$

where, $[p,r] = \{q \in R^d | p \le q \wedge q \le r\}$.

6 PROBLEM AND ITS IMPLEMENTATION

The developed DBB-MOPSO algorithm is applied to a pavement maintenance decision optimisation 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 indirect representations as shown in Figure 4. The pavement segment data denoted by AADT and X_2, X_4 and X_5, required for Equations (4) and (5), are given in Table 2 and Table 3**Error! Reference source not found.**. In addition, two other GA algorithms are used for comparison. In this regard, the proposed algorithm is compared with the above as they are the state which are two other GA based methods, namely, multi-objective genetic algorithm, and non-dominated sorting genetic algorithm II (NSGA-II) [22]. Like the PSO based algorithms, the two GA methods are also implemented in MATLAB. The encoding technique that is used to handle discrete variables for GA and NSGA-II is the direct representation, while combination of direct and indirect representation is applied for DBB-MOPSO and DMOPSO. Table 4 shows the settings with which the four evolutionary algorithms are run. Parameters in Table 4 are chosen that the evolutionary parameters in each of the algorithms are identical. Lower and upper bounds mean search domain of decision variables and included in table 4. Since our decision variables are between 1 and 5, the upper and lower bounds are set to these limits, so the variable is allowed to all the possible values.

Figure 4: Maintenance optimization problem.

Table 2: Pavement data for 5 sections over years 1-5.

Table 3: Pavement data for 5 sections over years 6-10.

Table 4: Description of the settings used for each algorithm.

The iterations for the PSO methods and the generations for the GA methods are set equal, although the concepts of iterations and generations, for the former and the latter respectively, do not have any similarities. It should be noted that out of the two PSO algorithms, the proposed method does not need the selection of any additional parameters, which is a clear advantage of it.

For the four algorithms, the performance metrics given in Section 5 are estimated. For verifying the non-dominated solutions spread in the entire region of the Pareto front, the diversity measure is estimated. Figure 5(a) shows the proposed algorithm has lower diversity at the 100th generation compared to the DMOPSO, NSGA-II and GA. Hence, the latter three perform better. However, the diversity values in the range of 0.95 – 1.00 between the algorithms is a good indication that all of them performed on a par. As shown in Figure 5(b), DBB- MOPSO has slightly smaller value of generational distance GD compared to DMOPSO and NSGA-II at the 100th iteration while it has significantly smaller value of GD compared to GA. Therefore, the convergence speed of the DBB-MOPSO to the Pareto front is better than the three algorithms at this stage.

Figure 5: The diversity metric (a) and generational distance (b) of the both algorithms.

According to Figure 6(a), the maximum spread of the DBB-MOPSO algorithm is approximately in the same range of DMOPSO, but the maximum spreads of the GA algorithm and NSGA-II over the whole iteration range are definitely greater than that of DBB-MOPSO. Also considering the Diversity measure, this means the GA and NSGA-II have better performance compared to DBB-MOPSO and DMOPSO. Figure 6(b) shows that DBB-MOPSO, DMOPSO, and NSGA-II at the first half of whole generations have approximately the same range of spacing but at the second half, the DBB-MOPSO and DMOPSO has smaller values of spacing. In addition, the GA has larger value of spacing at whole generations compared to DBB-MOPSO. This shows that the solutions of DBB-MOPSO and the DMOPSO are more uniformly spaced compared to GA and NSGA-II algorithms. The four metrics show that the GA algorithms outperform the PSO algorithms generally. Furthermore, it is also shown that DBB-MOPSO and DMOPSO have similar performances. This shows that there is no performance degradation when using the proposed PSO method instead of the existing one.

Figure 6: Maximum spread (a) and spacing (b) metrics of the both algorithms.

In order to show the distributions of solutions on the found Pareto fronts, Figure 7 presents the results from DBB-MOPSO and the other three algorithms. After 100 generations, for the DBB-MOPSO algorithm, 10 non-dominated solutions from 100 solutions are found as shown in figure 7. For the DMOPSO algorithm, there are 17 non-dominated solutions from the 100 solutions found. For the genetic based algorithms, the number of non-dominated solutions is much higher after 100 generations, as shown in Figure 7**Error! Reference source not found.**. Based on Figure 7, compared to the spread of non-dominated solutions of NSGA-II and GA, the non-dominated solutions spread of DBB-MOPSO is slightly better than that of DMOPSO, especially for the cost objective f_1.

Figure 7: Pareto fronts created by the four algorithms, together with the respective compromise solutions (in red).

To simulate the agency preferences, the compromise solution, as described in Section 4.7, is calculated for the Pareto fronts of the four algorithms. The solution having the maximum membership value ($\boldsymbol{\mu^i}$) in the archive is selected as the optimal pavement maintenance in both algorithms. In Figure 7**Error! Reference source not found.**, the found compromise solutions are plotted in red. Table 5 shows the optimal maintenance decisions, i.e. the compromise solutions, obtained from the Pareto fronts of DBB-MOPSO and DMOPSO. From Figure 7 and Table 5, it is clear that the Pareto fronts from the GA based algorithms are better than those obtained by the PSO ones. When compared to the latter the former show about 20 times increase in pavement performance. Moreover, NSGA-II has outperformed all the others. Although the overall values of pavement conditions found by GA and NSGA-II are better than by that found by DBB-MOPSO, the overall value of the maintenance cost found by DBB-MOPSO algorithm is better than that found by DMOPSO and GA by about 16% and 97% respectively. Furthermore, the Pareto fronts of DBB-MOPSO and DMOPSO are comparatively similar. However, for the compromised solutions of these two techniques are compared, the cost obtained by DBB-MOPSO is, as mentioned earlier, about 16% less. The difference between the two compromised solutions for the second objective of pavement performance is less than 1%. Hence, the proposed algorithm, DBB-MOPSO, can be said to outperform DMOPSO. Moreover, in the optimal maintenance plan found by DBB-MOPSO algorithm, given in table 4, there is heavier investment in the pavement all the others. Although the overall values of pavement conditions found by GA and NSGA-II are better than by that found by DBB-MOPSO, the overall value of the maintenance cost found by DBB-MOPSO algorithm is better than that found by DMOPSO and GA by about 16% and 97% respectively.

Furthermore, the Pareto fronts of DBB-MOPSO and DMOPSO are comparatively similar. However, for the compromised solutions of these two techniques are compared, the cost obtained by DBB-MOPSO is, as mentioned earlier, about 16% less. The difference between the two compromised solutions for the second objective of pavement performance is less than 1%. Hence, the proposed algorithm, DBB-MOPSO, can be said to outperform DMOPSO. Moreover, in the optimal maintenance plan found by DBB-MOPSO algorithm, given in Table 4, 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 the DMOPSO algorithm as shown in Table 5, there is heavy maintenance investment for most sections in the middle years.

Table 5: The pavement maintenance program based on the compromised solutions of the four algorithms.

Figure 8**Error! Reference source not found.** shows the time elapsed for each algorithm for the 100 generations. All algorithms are run on a PC with an Intel core i3 CPU (2.40 GHz) and a 6 GB RAM memory. Based on Figure 8, DBB-MOPSO consumes the least time of all, although the mechanisms of GA and PSO are not matched for a direct comparison. However, when compared to the other PSO method, DMOPSO, the proposed algorithm takes about 40% of time. Hence, this is a huge gain, especially for large sized optimizations problems, encountered in pavement maintenance scheduling. GA and NSGA take slightly longer than DBB-MOPSO for the same number of generations and iterations and identical amount of swarm and population.

Figure 8: Execution time (hours) for four algorithms.

This section covers the computational results found over four benchmark test functions. Actually, the DBB-MOPSO is developed and applied to solve pavement maintenance decision optimization problem that is unconstrained problem. Therefore, these test functions that are unconstrained optimization problems are selected. In addition, to examine the capability of developed algorithm in solving optimization problems that have different levels of difficulty, the dimensionalities of the chosen benchmark functions are differentiated. The different Pareto solutions are obtained from a benchmark website [79]. Several of standard test problems were selected for the experimental purpose. There functions are Fonseca and Fleming [80], Poloni [81], Schaffer N.1, Schaffer N.2 [82], and CEC 2009 [83]; these are defined in Table 6.

Table 6: Test problems.

The comparisons among different algorithms are made on the basis of Pareto fronts achieved at each benchmark function. Different figures are plotted for the found Pareto fronts to present clearly the quality of solutions. The Pareto fronts for Fonseca and Fleming's, Poloni's, Schaffer N.1, and Schaffer N.2 functions for four optimization algorithms are plotted (see Figure 9, Figure 10, Figure 11, Figure 12 and Figure 13). The performances of all the algorithms are competitively decent over this function, and the Pareto fronts found from these algorithms are uniformly distributed. However, in all benchmark test functions, the DBB-MOPSO and DMOPSO converge to the optimal Pareto front in only 5 out of 100 runs, while the NSGA II and GA converge to the Pareto front after 500 test runs. Based on the results of algorithm comparison and test problems, there is the significant difference in the pavement performance (objective f1) between DBB-MOPSO and DMOPSO with NSGA-II and GA, while there is no significant difference in four benchmark tests. A possible reason behind this is that since GA and NSGAII are very sensitive to many parameters such as population size, mutation operator, crossover operator, so on. Therefore, if there are too few chromosomes, the both algorithms have a few possibilities to execute crossover and only a small part of search space is explored.

Figure 9: Fonseca and Fleming function.

Figure 10: Poloni function.

Figure 11: Schaffer N.1 function.

Figure 12: Schaffer N.2 function.

Figure 13: CEC 2009 unconstrained problem 4.

From Table 7, the best results with respect to the IGD obtained by GA for all test functions except Fonseca and Fleming's function and CEC2009 problem2. However, overall, this table shows that the results with respect to the IGD metric for all test functions found by the DBB-MOPSO algorithm are fairly good compared to three algorithms. In addition, as shown in Table 7, the results with respect to the HV measure found by DBB-MOPSO algorithm for all benchmark functions are the best when compared to those obtained by all the other algorithms. Figure 14 and Figure 15 show the variation of IGD and HV results from the four algorithms with generations for all test functions. The DBB-MOPSO and DMOPSO find the best results with respect to the IGD and HV from the first five generations for all test functions. These results are still the same with generations, whereas the IGD and HV results obtained by GA and NSGAII are variable with generations. Therefore, the DBB-MOPSO and DMOPOS are faster to reach the true Pareto front than GA and NSGAII. Refer to Figure 14 and 15, DBB-MOPSO and DMOPSO algorithms reach the Pareto front quickly compared to the results of NSGA II and GA. This is because the performance of NSGA II and GA is sensitive to number of populations and number of generations. Unlike PSO algorithm, to improve their search capabilities, they need to explore huge search space efficiently.

Table 7: The results of IGD and HV for four test functions.

Figure 14: IGD for each algorithm for four test functions.

Figure 15: HV for each algorithm for four test functions.

7 CONCLUSIONS

A novel particle swarm algorithm is developed for a discrete multi-objective problem. This novel algorithm, being based on the bare-bones method for continuous search domains, is parameter free presenting a clear advantage over the algorithms where the user has to do parameter selection. To achieve usability toward discrete problems, the standard barebones algorithm is modified to incorporate velocity terms as in the standard PSO. The proposed algorithm is applied to find optimal rehabilitation scheduling considering the two objectives, namely the minimisation of the total pavement rehabilitation cost and the minimisation of the sum of all residual PCI values.

The formulation of a pavement maintenance problem is presented. Furthermore, the algorithm is applied to the problem along with three other algorithms: a particle swarm optimization method called DMOPSO, and two genetic techniques, GA and NSGA, namely. Implementation is in MATLAB. As this a multi-objective problem, non-dominated solutions obtained via the four techniques are benchmarked against each other, via their so-called compromised solutions. Here, the results show that the GA-based algorithms outperform both the PSO algorithms, for the same order of run-times, in terms of optimal objectives derived. For the same number of swarm iterations, it has also been shown that the objective function values obtained via the proposed method are better than those from the DMOPSO procedure for the same number of iterations. Especially, the cost resulting from the proposed DBB-MOPSO is about 8% lower than that from DMOPSO. Various standard performance measures are considered for the non-dominated solutions obtained via all four methods; these values are also traced across different iterations (for the PSOs) and generations (for the GAs). The measures indicate that the proposed method very similar to the DMOPSO in generating a Pareto front. Furthermore, the GA techniques outperform the PSO algorithms for most of the metrics.

As the time taken for optimization is a bottleneck, especially for large pavement networks, the run-time of the algorithms are also evaluated for the four methods tested. Although there is no basis to directly compare the time for the same number of iterations (for the PSOs) and generations (of the GAs), for the solutions described above, the proposed algorithm takes the least of all. The time consumed by DBB-MOPSO to produce the better solutions, as mentioned earlier, is about 40% of that taken by the DMOPSO method. This is a clear advantage of the proposed method, suggesting its suitability for large-scale pavement network scheduling.

In future, the proposed algorithm will be put through more validation by benchmarking. Especially, it is being tested for larger sized pavement networks. Furthermore, its different alterations will be made to the proposed updating scheme of particle velocities and positions and their effect on the solution quality will be examined. In addition, some pavement scheduling problems are formulated as constrained optimization problems. Some of these will also be tackled by the current proposal. Problems with more than two objectives are also planned to be tackled by the proposed optimization algorithm.

8 REFERENCES

[1] C. Chen, G.W. Flintsch, I.L. Al-Qadi, Fuzzy Logic-Based Life-Cycle Costs Analysis Model for Pavement and Asset Management, in: 6th Int. Conf. Manag. Pavements, TRB, Brisbane Queensland, Australia, 2004.

[2] A. Alsherri, K.P. George, Reliability Model for Pavement Performance, J. Transp. Eng. 114 (1988) 294–306.

[3] R. Haas, W.R. Hudson, J.P. Zaniewski, Modern Pavement Management, 2nd ed., Krieger Pub. Co., Cornell University, 1994.

[4] T.F. Fwa, W.F. Chan, K.Z. Hoque, Multi-Objective Optimisation for Pavement Maintenance Programming, J. Transp. Eng. 126 (2000) 367–374.

[5] K.A. Abaza, Iterative Linear Approach for Nonlinear Nonhomogenous Stochastic Pavement Management Models, J. Transp. Eng. 132 (2006) 244–256.

[6] W. Gu, Y. Ouyang, S. Madanat, Joint Optimisation of Pavement Maintenance and Resurfacing Planning, Transp. Res. Part B Methodol. 46 (2012) 511–519. doi:10.1016/j.trb.2011.12.002.

[7] P.L. Durango-Cohen, P. Sarutipand, Maintenance Optimisation for Transportation Systems with Demand Responsiveness, Transp. Res. Part C Emerg. Technol. 17 (2009) 337–348. doi:10.1016/j.trc.2009.01.001.

[8] Z. Wu, G.W. Flintsch, T. Chowdhury, Hybrid Multiobjective Optimisation Model for Regional Pavement-Preservation Resource Allocation, Transp. Res. Rec. J. Transp. Res. Board. 2084 (2008) 28–37. doi:10.3141/2084-04.

[9] A.A. Butt, M.Y. Shahin, S.H. Carpenter, J. V. Carnahan, Application of Markov Process to Pavement Management Systems at Network Level, in: 3rd Int. Conf. Manag. Pavements, TRB, San Antonio Texas, 1994: pp. 159–172.

[10] Y. Yin, S. Lawphongpanich, Y. Lou, Estimating investment requirement for maintaining and improving highway systems, Transp. Res. Part C Emerg. Technol. 16 (2008) 199–211. doi:10.1016/j.trc.2007.07.004.

[11] J. Farhan, T.F. Fwa, Incorporating Priority Preferences into Pavement Maintenance Programming, J. Transp. Eng. 138 (2012) 714–722. doi:10.1061/(ASCE)TE.1943-5436.0000372.

[12] C.U. Chikezie, A.T. Olowosulu, O.S. Abejide, Multiobjective optimization for pavement maintenance and rehabilitation programming using genetic algorithms, 5 (2013) 76–83.

[13] Y. Wang, S. Goldschmidt, Optimisation of Pavement Maintenance and Renewal Activities, in: 30th Conf. Aust. Institutes Transp. Res., University of Western Australia, Crawley, Perth, Western Australia, 2008: pp. 1–21.

[14] Y. Shen, Y. Bu, M. Yuan, A Novel Chaos Particle Swarm Optimisation (PSO) and its Application in Pavement Maintenance Decision, in: 4th IEEE Conf. Ind. Electron. Appl., IEEE, Xi'an China, 2009: pp. 3521–3526.

[15] N.R. Tayebi, F. Moghadasnejhad, A. Hassani, Analysis of Pavement Management Activities Programming by

Particle Swarm Optimisation, in: Int. Conf. Adv. Electr. Electron., ACEEE, Trivandrum, Kerala, India, 2010: pp. 149–154.

[16] T.F. Fwa, W.T. Chan, C.Y. Tan, Genetic-Algorithm Programing of Road Maintenance and Rehabilitation, J. Transp. Eng. 122 (1996) 246–253.

[17] J.-S. Chou, T.-S. Le, Reliability-Based Performance Simulation for Optimized Pavement Maintenance, Reliab. Eng. Syst. Saf. 96 (2011) 1402–1410. doi:10.1016/j ress.2011.05.005.

[18] A.V. Moreira, J. Oliveira, L. Costa, T.F. Fwa, Assessment of different genetic algorithms for pavement management systems, in: 8th Int. Conf. Maint. Rehabil. Pavements, Singapore, 2016: pp. 462–472. http://hdl handle net/1822/43192.

[19] J. Santos, A. Ferreira, G. Flintsch, An adaptive hybrid genetic algorithm for pavement management, Int. J. Pavement Eng. 8436 (2017) 1–21. doi:10.1080/10298436.2017.1293260.

[20] J.L. Zavitski, R.R. Piane, 30 Years - 20 State DOTs – Trends in Pavement Management Observed Through Real World Implementation at the State DOT Level., in: 9th Int. Conf. Manag. Pavement Assets, Alexandria, Virginia, USA, 2015: pp. 1–11.

[21] Y. Zhang, D.-W. Gong, Z. Ding, A Bare-Bones Multi-Objective Particle Swarm Optimisation Algorithm for Environmental/Economic Dispatch, Inf. Sci. (Ny). 192 (2012) 213–227. doi:10.1016/j.ins.2011.06.004.

[22] K. Deb, A. Member, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm :, 6 (2002) 182–197.

[23] K. Deb, H. Jain, An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints, Ieeexplore.Ieee.Org. 18 (2013) 1–1. doi:10.1109/TEVC.2013.2281534.

[24] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Trans. Evol. Comput. 11 (2007) 712–731. doi:10.1109/TEVC.2007.892759.

[25] S.Pal, B.Y. Qu, S.Das, P.N. Suganthan, Optimal Synthesis of Linear Antenna Ar- Rays With Multi-Objective Differential Evolution, Prog. Electromagn. Res. B. 21 (2010) 87–111.

[26] Hui Li, Qingfu Zhang, Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2009) 284–302. doi:10.1109/TEVC.2008.925798.

[27] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition, Tech. Rep. (2009) 1–30.

[28] P.C. Chang, S.H. Chen, Q. Zhang, J.L. Lin, MOEA/D for flowshop scheduling problems, 2008 IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell. (2008) 1433–1438. doi:10.1109/CEC.2008.4630982.

[29] Chih-Ming Chen, Ying-ping Chen, Tzu-Ching Shen, John Kar-Kin Zao, Optimizing degree distributions in LT codes by using the multiobjective evolutionary algorithm based on decomposition, Evol. Comput. (CEC), 2010 IEEE Congr. (2010). doi:10.1109/CEC.2010.5586340.

[30] A. Konstantinidis, C. Charalambous, A. Zhou, Q. Zhang, Multi-objective Mobile Agent-based Sensor Network Routing using MOEA / D, IEEE World Congr. Comput. Intell. (2010) 18–23.

[31] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA_D on CEC09 unconstrained MOP test instances.pdf, (2009) 203–208.

[32] Q. Jiang, L. Wang, X. Hei, G. Yu, Y. Lin, X. Lu, MOEA/D-ARA+SBX: A New Multi-Objective Evolutionary Algorithm Based on Decomposition with Artificial raindrop algorithm and Simulated Binary Crossover, Knowledge-Based Syst. 107 (2016) 197–218. doi:10.1016/j.knosys.2016.06.007.

[33] L. Wang, C. Singh, Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm, Electr. Power Syst. Res. 77 (2007) 1654–1664. doi:10.1016/j.epsr.2006.11.012.

[34] A. Moraglio, Multiobjective Optimal Power Flow Using Hybrid Evolutionary Algorithm, Int. J. Electr. Comput. Eng. 4 (2010) 591–596.

[35] A. Bhattacharya, P.K. Chattopadhyay, Hybrid differential evolution with biogeography-based optimization algorithm for solution of economic emission load dispatch problems, Expert Syst. Appl. 38 (2011) 14001–14010. doi:10.1016/j.eswa.2011.04.208.

[36] X. Wei, S. Fujimura, Multi-Objective Quantum Evolutionary Algorithm for Discrete Multi-Objective Combinational Problem, (2010). doi:10.1109/TAAI.2010.18.

[37] L. Wang, Q.-K. Pan, P.N. Suganthan, W.-H. Wang, Y.-M. Wang, A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems, Comput. Oper. Res. 37 (2010) 509–520. doi:10.1016/j.cor.2008.12.004.

[38] M.F. Tasgetiren, P.N. Suganthan, Q. Pan, An ensemble of discrete differential evolution algorithms for solving the generalized traveling salesman problem, Appl. Math. Comput. 215 (2010) 3356–3368. doi:10.1016/j.amc.2009.10.027.

[39] T.X.C.C.S. Chong, Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives, (2014). doi:10.1007/s10845-014-0869-8.

[40] S.N. Deepa, G. Sugumaran, Model order formulation of a multivariable discrete system using a modified particle swarm optimization approach, Swarm Evol. Comput. 1 (2011) 204–212. doi:10.1016/j.swevo.2011.06.005.

[41] H.P. Hsu, A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously, Swarm Evol. Comput. 27 (2016) 156–168. doi:10.1016/j.swevo.2015.11.002.

[42] M.F. Tasgetiren, Q. Pan, P.N. Suganthan, A.H. Chen, A Discrete Artificial Bee Colony Algorithm for the Permutation Flow Shop Scheduling Problem with Total Flowtime Criterion, (2010).

[43] Q.-K. Pan, M. Fatih Tasgetiren, P.N. Suganthan, T.J. Chua, A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, Inf. Sci. (Ny). 181 (2011) 2455–2468.

doi:10.1016/j.ins.2009.12.025.

[44] A. Kalam, A. Maria, A.C. Rocha, E.M.G.P. Fernandes, Improved binary arti fi cial fi sh swarm algorithm for the 0 – 1 multidimensional knapsack problems, 14 (2014) 66–75. doi:10.1016/j.swevo.2013.09.002.

[45] M. Metlicka, D. Davendra, Chaos Driven Discrete Artificial Bee Algorithm for Location and Assignment Optimisation Problems, Swarm Evol. Comput. (2015) 1–14. doi:10.1016/j.swevo.2015.03.002.

[46] E. Fatnassi, O. Chebbi, J. Chaouachi, Discrete honeybee mating optimization algorithm for the routing of battery-operated automated guidance electric vehicles in personal rapid transit systems, Swarm Evol. Comput. 26 (2016) 35–49. doi:10.1016/j.swevo.2015.08.001.

[47] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (2016) 1053–1073. doi:10.1007/s00521-015-1920-1.

[48] L. Kumar, S.R. K, A. Verma, B.K. Panigrahi, R. Kumar, Binary Grey Wolf Optimizer for large scale unit commitment problem, Swarm Evol. Comput. (2017) 0–1. doi:10.1016/j.swevo.2017.08.002.

[49] S. Mirjalili, P. Jangir, S. Saremi, Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems, Appl. Intell. 46 (2017) 79–95. doi:10.1007/s10489-016-0825-8.

[50] C. Lu, X. Li, L. Gao, W. Liao, J. Yi, An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times, Comput. Ind. Eng. 104 (2017) 156–174. doi:10.1016/j.cie.2016.12.020.

[51] C. Chen, Soft Computing-based Life-Cycle Cost Analysis Tools for Transportation Infrastructure Management, Virginia Polytechnic Institute and State University, 2007.

[52] FHWA, Life-Cycle Cost Analysis in Pavement Design, Washington, DC, 1998.

[53] F. Javed, Integrated Prioritisation and Optimisation Approach for Pavement Management, National University of Singapore, 2011.

[54] M.Y. Shahin, Pavement Management for Airports, Roads and Parking Lots.., 2nd ed., Springer, New York, NY, USA, 2005.

[55] P. Herabat, A. Tangphaisankun, Multi-Objective Optimisation Model using Constraint-Based Genetic Algorithms for Thailand Pavement Management, J. East. Asia Soc. Transp. Stud. 6 (2005) 1137–1152.

[56] M. Mahmood, Cross Validation of Multi Input Deterioration Prediction Model (MID-PM) for Nnetwork Level, in: Inaug. Coll. Art, Des. Built Environ. Dr. Conf., Nottingham Trent University, Nottingham, UK, 2014: pp. 44–52.

[57] S.S. Rao, Engineering Optimisation: Theory and Practice, 4th ed., John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.

[58] D. Teodorović, Swarm Intelligence Systems for Transportation Engineering: Principles and Applications, Transp. Res. Part C Emerg. Technol. 16 (2008) 651–667. doi:10.1016/j.trc.2008.03.002.

[59] M. Pluhacek, R. Senkerik, D. Davendra, Chaos particle swarm optimization with Eensemble of chaotic systems, Swarm Evol. Comput. 25 (2015) 29–35. doi:10.1016/j.swevo.2015.10.008.

[60] Y. Shi, R.C. Eberhart, Parameter Selection in Particle Swarm Adaptation, in: Int. Conf. Evol. Program. VII, Springer-Verlag, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Berlin, Germany, 1998: pp. 591–600.

[61] A.B. de Carvalho, A. Pozo, Measuring the Convergence and Diversity of CDAS Multi-Objective Particle Swarm Optimisation Algorithms: A Study of Many-Objective Problems, Neurocomputing. 75 (2012) 43–51. doi:10.1016/j.neucom.2011.03.053.

[62] T. Navalertporn, N. V. Afzulpurkar, Optimization of tile manufacturing process using particle swarm optimization, Swarm Evol. Comput. 1 (2011) 97–109. doi:10.1016/j.swevo.2011.05.003.

[63] A. Osyczka, An Approach to Multicriterion Optimisation Problems for Engineering Design, Comput. Methods Appl. Mech. Eng. 15 (1978) 309–333.

[64] A.P. Engelbrecht, Computational Intelligence: An Introduction, 2nd ed., John Wiley & Sons, 2007.

[65] C.-J. Liao, C.-T. Tseng, P. Luarn, A Discrete Version of Particle Swarm Optimisation for Flowshop Scheduling Problems, Comput. Oper. Res. 34 (2007) 3099–3111. doi:10.1016/j.cor.2005.11.017.

[66] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization, Swarm Evol. Comput. 9 (2013) 1–14. doi:10.1016/j.swevo.2012.09.002.

[67] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Int. Conf. Syst. Man, Cybern. Comput. Cybern. Simul., IEEE, Orlando, FL, 1997: pp. 4104–4108.

[68] H. Izakian, B.T. Ladani, A. Abraham, A Discrete Particle Swarm Optimisation Approach for Grid Job Scheduling, Int. J. Innov. Comput. Inf. Control. 6 (2010) 1–9.

[69] S. Mostaghim, J. Teich, Strategies for Finding Good Local Guides in Multi-Objective Particle Swarm Optimisation (MOPSO), in: Swarm Intell. Symp., IEEE, 2003: pp. 26–33. doi:10.1109/SIS.2003.1202243.

[70] D.R.C. Silva, C.J.A. Bastos-filho, A Multi-Objective Particle Swarm Optimizer Based on Diversity, in: Second Int. Conf. Intell. Syst. Appl., IARIA, Venice, Italy, 2013: pp. 109–114.

[71] S. Mirjalili, A. Lewis, Novel performance metrics for robust multi-objective optimization algorithms, Swarm Evol. Comput. 21 (2015) 1–23. doi:10.1016/j.swevo.2014.10.005.

[72] R. a. Santana, M.R. Pontes, C.J. a. Bastos-Filho, A Multiple Objective Particle Swarm Optimisation Approach Using Crowding Distance and Roulette Wheel, in: Ninth Int. Conf. Intell. Syst. Des. Appl., IEEE, Pisa, 2009: pp. 237–242. doi:10.1109/ISDA.2009.73.

[73] M. Salazar-Lechuga, J.E. Rowe, Particle Swarm Optimisation and Fitness Sharing to solve Multi-Objective Optimisation Problems, in: Congr. Evol. Comput., IEEE, 2005: pp. 1204–1211. doi:10.1109/CEC.2005.1554827.

[74] S.-J. Tsai, T.-Y. Sun, C.-C. Liu, S.-T. Hsieh, W.-C. Wu, S.-Y. Chiu, An Improved Multi-Objective Particle Swarm Optimizer for Multi-Objective Problems, Expert Syst. Appl. 37 (2010) 5872–5886. doi:10.1016/j.eswa.2010.02.018.

[75] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24. doi:10.1016/j.swevo.2015.05.002.

[76] M.R. Sierra, C.A. Coello Coello, Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and E-Dominance, Lect. Notes Comput. Sci. 3410 (2005) 505–519. doi:10.1007/978-3-540-31880-4_35.

[77] A.P. Guerreiro, C.M. Fonseca, M.T. Emmerich, A Fast Dimension-Sweep Algorithm for the Hypervolume Indicator in Four Dimensions., in: 24th Can. Conf. Comput. Geom., Charlottetown, Prince Edward Island, Canada, 2012: pp. 77–82. doi:http://2012.cccg.ca/papers/paper70.pdf.

[78] N.C. Sahoo, S. Ganguly, D. Das, Multi-objective planning of electrical distribution systems incorporating sectionalizing switches and tie-lines using particle swarm optimization, Swarm Evol. Comput. 3 (2012) 15–32. doi:10.1016/j.swevo.2011.11.002.

[79] J. J. Durillo, A.J. Nebro, jMetal Web site. [online] Jmetal.sourceforge.net., (2008). http://jmetal.sourceforge.net/problems html (accessed January 28, 2018).

[80] C.M. Fonseca, P.J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms II: Application example, 1995.

[81] C. Poloni, Hybrid GA for multi objective aerodynamic shape optimization, Genet. Algorithms Eng. Comput. Sci. 33 (1995) 397–415.

[82] J.D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: 1st Int. Conf. Genet. Algorithms, 1985: pp. 93–100. http://dl.acm.org/citation.cfm?id=657079.

[83] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition, Tech. Rep. (2009) 1–30.

Table 1

Rehabilitation options in consideration.

NO.	Treatment action
1	Do nothing
2	AC* overlay 1 inches
3	AC overlay 2 inches
4	AC overlay 4 inches
5	AC overlay 6 inches

*Asphalt Concrete

Table 2

Pavement data for 5 sections over years 1-5.

Years	1					2					3					4					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
Cracking length	232 9	586 9	1 4	0	0	252 4	589 5	11 3	0	0 8	323 4	585 7	66 3	13 8	22 7	249	579 1	35 2	4 95	68 8	0	588 9	38 5	0	98 8
Cracking area	15 6	0	0	0	0	12 1	1 57	0	0	0	0 8	2 62	1 15	0	0	28 3	13 8	12 32	0	0	0	13	22 7	0	0
Age	7 26	14 92	8 61	2 08	5 02	8 35	16 01	9 67	3 16	6 02	9 31	16 97	10 64	4 02	7 09	10 42	18 08	11 75	5 02	8 09	11 42	19 08	12 75	5 58	8 74
AADT	3674	4325	9140	2054	2880	5820	2975	9362	2456	2949	5830	3067	9608	2485	3000	6068	3163	9854	2515	3063	6306	3227	10100	2329	3123

Table 3

Pavement data for 5 sections over years 6-10.

Years	6					7					8					9					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
Cracking length	370 7	598 7	41 4	1	64 8	138 2	608 5	25	3 5	94 1	718 4	159 1	131 5	16 7	169 9	457 2	221 1	108 2	23 4	172 8	457 2	261 4	111 7	27 9	171 9
Cracking area	47	12 1	31 8	0	73 9	522 1	11 2	117 5	0	64 3	144 9	122 1	66 9	0	31 8	92	111 4	117 2	0	73 2	92	104 5	73 6	0 8	66 6
Age	12 03	20 08	13 62	7 09	9 81	13 06	21 15	14 41	7 89	10 74	14 44	22 14	15 87	8 97	12 21	15 36	23 14	16 8	9 97	13 23	16 36	23 79	17 8	10 65	13 85
AADT	6558	3291	10570	2143	3183	6831	3355	11270	1957	3243	7104	3415	11800	1771	3303	7377	3517	12100	1847	3363	7672	3615	12500	1999	3423

Table 4

Description of the settings used for each algorithm

Algorithm	Parameters
DBB-MOPSO	Swarm size = 100; dimensions = 50; archive size of 100; number of iterations = 100.
DMOPSO	Swarm size = 100; dimensions = 50; archive size of 100; number of iterations = 100; a velocity range [6, -6] [67], c_1 = 2, c_2 = 2 [68].
GA	Population size = 100; number of decision variables = 50; number of generations = 100; lower bound =1; upper bound = 5;
NSGA-II	Population size = 100; number of decision variables = 50; number of generations = 100; lower bound =1; upper bound = 5;

Table 5

The pavement maintenance program based on the compromised solutions of the four algorithms.

Years		1					2					3					4					5					6					7					8					9					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	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Decisions	DBB-MOPSO	5	5	4	3	3	4	4	5	4	4	5	1	4	3	3	4	1	5	2	3	5	5	4	5	5	2	3	4	3	5	1	2	4	5	5	1	1	3	2	2	4	1	4	1	1	3	1	3	2	4
	DMOPSO	2	4	5	1	5	4	5	4	2	5	5	4	5	2	5	2	5	1	5	2	3	5	5	4	5	4	5	5	2	2	3	2	4	4	4	4	3	4	2	1	4	2	4	2	3	2	1	1	1	2
	GA	4	5	5	5	4	5	4	4	4	4	4	4	5	5	4	5	4	5	4	5	5	4	5	4	4	4	5	5	4	5	5	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	NSGA-II	5	4	5	4	5	5	2	5	3	5	4	1	5	4	2	4	4	5	4	4	4	2	5	4	2	4	2	5	2	1	2	2	5	2	2	4	1	5	1	2	2	1	4	1	1	1	1	4	1	1

Table 6

Test problems.

Problem Name	Description	n	Variable Bounds (search domain)	Number of true Pareto solutions
Fonseca and Fleming function	$f_1(x) = 1 - exp\left[-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right]$ $f_2(x) = 1 - exp\left[-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right]$	3	$-4 \le x_i \le 4,$ $1 \le i \le n$	434
Poloni's function	$f_1(x,y) = \left[1 + \left(A_1 - B_1(x,y)\right)^2 + \left(A_2 - B_2(x,y)\right)^2\right]$ $f_2(x,y) = (x+3)^2 + (y+1)^2$ $where = \begin{cases} A_1 = 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2) \\ A_2 = 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2) \\ B_1(x,y) = 0.5\sin(x) - 2\cos(x) + \sin(y) - 1.5\cos(y) \\ B_2(x,y) = 1.5\sin(x) - \cos(x) + 2\sin(y) - 0.5\cos(y) \end{cases}$	2	$-\pi \le x, y \le \pi$	113
Schaffer function N. 1	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	1	$10 \le x \le 10$	201
Schaffer function N. 2	$f_1(x) = \begin{cases} -x, & if\ x \le 1 \\ x - 2, & if\ 1 < x \le 3 \\ 4 - x, & if\ 3 < x \le 4 \\ x - 4, & if\ x > 4 \end{cases}$ $f_2(x) = (x-5)^2$	1	$-5 \le x \le 10$	119
CEC 2009 unconstrained problem 1	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert}\sum_{j \in J_1}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2 \rvert}\sum_{j \in J_2}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ Where J_1={j/j is odd and $2 \le j \le n$} and J_2={j/j is even and $2 \le j \le n$}	30	$0 \le x_1 \le 1$ $-1 \le x_j \le 1$	1000
CEC 2009 unconstrained problem 2	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert}\sum_{j \in J_1} {y_j}^2, \quad f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2 \rvert}\sum_{j \in J_2} {y_j}^2$ Where J_1={j/j is odd and $2 \le j \le n$} and J_2={j/j is even and $2 \le j \le n$}	30	$0 \le x_1 \le 1$ $-1 \le x_j \le 1$	1000

	$y_j = x_j - \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right) \quad j\epsilon J_1$ $y_j = x_j - \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) \quad j\epsilon J_2$			
CEC 2009 unconstrained problem 4	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h(y_j), \quad f_2 = 1 - {x_1}^2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} h(y_j)$ Where J_1={j/j is odd and $2 \leq j \leq n$} and J_2={j/j is even and $2 \leq j \leq n$} $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \ j = 2, \ldots., n$ $h(t) = \frac{\lvert t\rvert}{1 + e^{2\lvert t\rvert}}$	30	$0 \leq x_1 \leq 1$ $-2 \leq x_j \leq 2$	1000
CEC 2009 unconstrained problem 7	$f_1 = + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} {y_j}^2, \quad f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} {y_j}^2$ Where J_1={j/j is odd and $2 \leq j \leq n$} and J_2={j/j is even and $2 \leq j \leq n$} $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \ j = 2, \ldots., n$	30	$0 \leq x_1 \leq 1$ $-1 \leq x_j \leq 1$	1000

Table 7

The results of IGD and HV for various test functions.

Algorithm	Description	Inverted Generational Distance (IGD)								Hypervolume (HV)							
		Fonseca & Fleming	Poloni	Schaffer N 1	Schaffer N 2	CEC2009 problem1	CEC2009 problem2	CEC2009 problem4	CEC2009 problem7	Fonseca & Fleming	Poloni	Schaffer N 1	Schaffer N 2	CEC2009 problem1	CEC2009 problem2	CEC2009 problem4	CEC2009 problem7
DBB-MOPSO	100pop 5gen	0.1803	29.3408	0.1154	15.5676	0.9798	1.3365	0.2340	1.0130	0.1962	0.8329	0.735	0.976	0.3843	0.4736	0.3121	0.1417
DMOPSO	100pop 5gen	0.1327	31.5803	0.1154	16.0385	1.2803	1.3461	1.4140	0.9883	0.1401	0.8389	0.7921	0.9771	0.3657	0.4511	0.3008	0.2219
GA	250 pop, 500gen	2.9841	0.9255	0.0399	8 9695	0.2400	1.3392	0.0452	0.4452	0.3029	0.8641	0.83	0.9779	0.5320	0.6252	0.3147	0.3957
NSGAII	250 pop, 500gen	1.3872	28.937	0.0387	16.0385	1.4325	0.0434	0.0165	1.4205	0.3106	0.8601	0.8329	0.9787	0.6669	0.6676	0.3302	0.4929