Application of chaos discrete particle swarm optimization algorithm on pavement maintenance scheduling problem

Kawther Ahmed, Belal Al-Khateeb & Maher Mahmood

Cluster Computing
The Journal of Networks, Software Tools and Applications

ISSN 1386-7857

Cluster Comput
DOI 10.1007/s10586-018-2239-3

Springer

Cluster Computing
https://doi.org/10.1007/s10586-018-2239-3

Application of chaos discrete particle swarm optimization algorithm on pavement maintenance scheduling problem

Kawther Ahmed[1] · Belal Al-Khateeb[1] · Maher Mahmood[2]

Received: 18 January 2018 / Accepted: 16 February 2018

Abstract

Particle swarm optimization (PSO) is one of the most popular and successful optimization algorithms used for solving single objective and multi-objective optimization problems. It is found that the Multi objective particle swarm optimization (MOPSO) has ability to find the optimal solution quickly and more efficient than other optimization algorithms. In this paper, a discrete (binary) MOPSO with chaos methods is developed and applied to pavement maintenance management. The main objective of this research is to find optimal maintenance and rehabilitation plan for flexible pavement with minimum maintenance cost and maximum pavement performance. This research is the first attempt to combine the crossover operation with velocity and position with multi objective PSO algorithm. The results show that the improvements in pavement performance and cost objectives are 94.65 and 54.01% respectively, while the improvement in execution time is 99.9%. In addition, it is found that the developed algorithm is able to converge to the optimal solution quickly, comparing with another PSO algorithm.

Keywords Particle swarm optimization · Pavement maintenance · Multi-objective optimization · Chaotic mapping · Binary PSO

1 Introduction

Roads are one of the most important capital resources for cities, hence it is necessary to be perfect to provide comfortability and safety for the road users. The pavement conditions depend on design, excavation control, construction, Maintenance and Rehabilitation (M&R) [1]. The functional and structural conditions of pavement are usually deteriorating with time due to various factors such as repeated adverse weather conditions, heavier loads, changing temperatures, traffic passage and poor reinstatement following excavation by public utility companies, these factors may be lead to potholes, cracking, and texture loss. In addition, all these factors lead to decrease the quality of pavement and cause serious problems. Therefore, the earlier maintenance will cause the less damage [2].

There are several researches used different computational intelligence methods for pavement maintenance decisions. In those researches, single and multi-objective problems were formulated and solved by different evolutionary algorithms such as Genetic Algorithm (GA) [3], K-Nearest Neighbor (K-NN) [4], Particle Swarm Optimization (PSO) and Ant Colony (AC) [5]. Fwa et al. [6] used GA to find optimal pavement maintenance activities at the network level. Elhadidy et al. [7] developed multi objective model using GA for programming maintenance actions. This model was implemented on expressway in Egypt by considering maximum Pavement Condition Index (PCI) and minimum maintenance costs. In recent years, researchers were started to use particle swarm optimization for pavement maintenance. Chou and Le [8] formulated

✉ Kawther Ahmed
kawther.ahmed2017@gmail.com

Belal Al-Khateeb
belal@computer-college.org

Maher Mahmood
maher.mahmood@uoanbar.edu.iq

1 Computer Science Department, College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq

2 Civil Engineering Department, Faculty of Engineering, University of Anbar, Ramadi, Iraq

Published online: 01 March 2018

MOPSO algorithm to find overlay layer thickness and the timing of maintenance labor. The maintenance cost and performance reliability of the pavement were considered simultaneously in the algorithm development as multi-objective problem. Mahmood [9] developed novel barebones particle swarm optimization for discrete optimization problems. This algorithm was applied to find optimal maintenance decisions for flexible pavement. Yi Shen et al. [10] proposed and applied chaos particle swarm optimization (CPSO) as single objective continuous problem to program pavement maintenance actions of expressway. It was found that CPSO has ability to find all optimal solutions than original PSO and Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Moreira et al. [11] used two optimization algorithms for pavement maintenance decisions. The first algorithm was a genetic algorithm to optimize a single-objective problem, whereas the second algorithm was NSGA-II for solving a multi-objective problem. Santos et al. [12] combined GA with Local Search (LS) methods to develop a novel Adaptive Hybrid Genetic Algorithm (AHGA) to find the optimal pavement maintenance strategy. The pavement maintenance scheduling problem is complex combinatorial optimization problem consisting of a large number of pavement segments and the associated treatment decision variables covering multiple time periods. The majority of previous researches could not consider all the optimization objectives for finding pavement maintenance plan. In addition, since the pavement maintenance scheduling problem is high dimensional optimization problem, the number of researchers has tried to applied different algorithms to address this problem with less time of execution and also avoid fall in local optima. In this paper, a novel discrete (binary) chaos is developed with multi objective PSO to solve the pavement maintenance scheduling problem. It has ability to address high dimensional pavement maintenance scheduling problem efficiently and converge to optimal solutions quickly with less execution time. In addition, it is the first attempt to combine the crossover operation with velocity and position in CPSO algorithm. This can improve the performance of the algorithm and then avoid it to fall in local optima hence leading to a significant improvement in results and execution time.

2 Pavement maintenance decision problem

2.1 Problem parameters

There are data and decision criteria used to perform maintenance and rehabilitation analysis. Those data and decision criteria are existing condition of the pavement based on distresses, minimum acceptable level of service, study period, maintenance cost and the budget. To estimate the maintenance needs, the highway network is divided into a number of pavement sections which have equal length [13].

Agency cost of highway network is the intervention required to design, build and invest a highway network. It comprises of the cost of maintenance, rehabilitation and reconstruction. Rehabilitation is required for highway network at least once in its lifetime to maintain it above the minimum acceptable service and safety level. The cost of any specific rehabilitation action comes from: primary engineering, materials and construction management. If a rehabilitation activity is to be implemented in subsequent years, then its cost can be deducted to the present value as the following:

$$\text{Present\_Cost} = \text{Future\_Cost} \times \text{PWF} \tag{1}$$

where, PWF is the present worth factor, given by:

$$PWF = \frac{1}{(1 + DR)^{ti}} \tag{2}$$

where DR is discount rates and the typical range of it recommended by federal highway administration is 3–5% [14], ti is the time at which the money is spent (specified in years).

Depending on the condition, highway authorities have the option to select rehabilitation action from an activities list of. Such list, which is also adopted in this research, is shown in Table 1. Furthermore, it is necessary to determine the trigger level for each maintenance activity. The trigger level is defined as the minimum level of pavement service ability, such that the maintenance must be implemented when the pavement section reaches it. The highway authority usually specifies the total length of the study period. Moreover, the length of the unit analysis period, that are commonly 1 year, is chosen based on the requirements of the highway agency [15].

Table 1 Pavement maintenance and rehabilitation strategies

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

AC asphalt concrete

2.2 Objective functions

Pavement maintenance systems have common objectives as defined by road authorities include: to minimize user costs by selecting and scheduling treatment procedures to minimize delay and traffic disruption, to minimize the present worth of overall treatment costs over the analysis period, and maintaining the pavement performance over the minimum acceptable level with resources available [16]. To make optimal investment decisions in the maintenance field, it is important to improve the M&R decision in view of multiple objectives such as maximum performance, minimum cost, etc. Therefore, the particle swarm optimization technique with multi-objective are used for scheduling pavement maintenance activities. Multi-objective programming for pavement maintenance and rehabilitation can be formulated mathematically as follows [9]:

Minimize the total pavement maintenance cost

$$f_1(x) = \sum_{d=1}^{D}\sum_{p=1}^{N}\sum_{m=1}^{M} x_{m,p,d} C_m L_p W_p (1+DR)^{-d} \tag{3}$$

Minimize the sum of all residual PCI values

$$f_2(x) = \sum_{d=1}^{D}\sum_{p=1}^{N}\sum_{m=1}^{M} x_{m,p,d}\left[(PCI_{\max} - PCI_{p,d}) L_p W_p AADT_{p,d}\right] \tag{4}$$

where,

$$x_{m,p,d} = \begin{cases} 1, & \textit{if treatment m fo } \text{section } \textit{p at time d is selected} \\ 0, & \textit{otherwise} \end{cases}$$

where d is any time in the analysis period, and D is the total analysis period (both are usually specified in years); N is the total number of pavement sections; m is the treatment type; M stands for the total number of treatment types; p is the pavement section number under consideration; L_p is the length of pavement section p; C is the unit cost of treatment type m; DR is the discount rate; W stands for the width of section p; PCI is the maximum PCI level (100%); PCI, d is the PCI for section p at time d; AADT, d is the annual average daily traffic for section p at time d.

2.3 Pavement condition prediction model

The Pavement Condition Index (PCI) is indicator to evaluate the overall conditions of pavement. It is based on visual survey results that are determined by distress type, quantity, and severity. Field verification of the inspection approach has proved that PCI is a good indicator of structural integrity and operational status. In addition, the PCI is a valuable indicator to determine both the current state and future performance under current traffic conditions. A pavement deterioration model is an vital stage when estimating maintenance needs, and when determining road user costs and benefits of the maintenance application [17]. Therefore, there is essential to prediction models that capable to predict pavement deterioration by considering distress, traffic loading, pavement age, and maintenance effects. For arterial roads in the wet freeze climatic region is used to estimate future pavement condition:

$$\mathrm{PCI} = 97.744 - 0.15X5 - 0.064X4 - 0.515X2 + 3.748X3 \tag{5}$$

where *PCI* is the pavement condition index*;* *X*5 is the cracking area (alligator, edge, and block); *X*2 is the pavement age; *X*3 is the maintenance effect (inlay and overlay thickness); *X*4 is the longitudinal and transverse cracking length [9].

3 Particle swarm optimization (PSO)

The PSO algorithm is one of computational algorithms inspired from animal such as birds or fish flocks and ant colonies. PSO is a population based search algorithm that was developed by Kennedy and Eberhart in 1995 [18]. The swarm consists of particles, each one representing a potential solution in optimization problem. These particles have two main attributes: position and velocity. The position of particle is updated according to its previous experiences and that of its neighbor. The velocity is adjustable to determine the direction that a particle need to move therefore improves its current position. During iterations, each particle i updates the velocity and position of its jth dimension at iteration t + 1 by using the following equations:

$$V_{i,j}(\mathrm{t}+1) = wV_{i,j}(\mathrm{t}) + r_1c_1\left[\mathrm{Pbest}_{i,j}(\mathrm{t}) - X_{i,j}(\mathrm{t})\right] + r_2c_2\left[Gbest(\mathrm{t}) - X_{i,j}(\mathrm{t})\right] \tag{6}$$

$$X_{i,j}(\mathrm{t}+1) = X_{i,j}(\mathrm{t}) + V_{i,j}(\mathrm{t}+1) \tag{7}$$

where Vi, j(t) is a velocity of particle i at iteration t; Xi, j(t) it is a position of i particle at iteration t and it depends on previous position and previous velocity; w is the inertia weight that is used to control the influence of the previous velocities on the current velocity [19]; r_1 and r_2 are two random number between (0,1); c_1 and c_2 are learning factors or acceleration factors that are fixed numbers; Pbest $\mathrm{i,j(t)}$ is the local best particle i in all swarm that have the smallest fitness value obtained so far in one iteration *t;* Gbest(t) is the leader of the swarm or global best position of all population, is the best one of the set of local best position.

The particle leader in each generation guides the particles to move towards the optimal positions. The performance of each particle in the swarm is evaluated according to objective function or the fitness function of the optimization problem [20, 21].

4 Multi-objective optimization problems

After the considerable successes achieved by the single objective problems, the researchers began to work on using PSO in multi objectives problems [19]. In multi-objective particle swarm optimization (MOPSO), there is no single optimal solution but there are a set of non-dominated solutions called Pareto-optimal solutions. A decent trade-off between solutions that represent an equalization between the objectives must be found. The Pareto optimality definition is that "A decision vector x^* ε F, is Pareto-optimal if there does not exist a decision vector, x ≠ x* ε F that dominates it, for a set of objective functions $\{f_1, f_2, \ldots f_k\}$ provided that the solution is possible x^* dominates another feasible solution x, then it is denoted by F(x*) < F(x) [15]. This function selects the non-dominated solutions and ignore the others. The difficulty with MOPSO algorithm is to find the leader for the swarm (the best global particle). Therefore, the researchers proposed to use archive to store all solution that found in the search space and select the leader from archive according to some sort of quality measure. The most common one is sigma method which can obtain non-dominated solutions with a very good diversity [22]. In this paper, the sigma method is employed to find the leader [23].

5 Discrete (binary) particle swarm optimization

The original PSO algorithm deals with continuous values. In 1997, Kennedy and Eberhart suggested a discrete (binary) model of PSO for solving discrete problems [24]. In the binary PSO, the global best and personal best of particles are updated as in continuous model, but the velocity and position are updated in different way. The position in binary PSO should be either 0 or 1. these values depend on velocity that will give a different value according to the following equations [25] [26]:

$$V_{i,j}(\mathrm{t}) = sig\left(V_{i,j}(\mathrm{t})\right) \tag{8}$$

$$sig\left(V_{i,j}(t)\right) = \frac{1}{1 + e^{-Vi,j(t)}} \tag{9}$$

where *sig* is the sigmoid function.

The position equation is changed to make the positions of all particles is only 0 or 1 according to the following equation:

$$X_{i,j}(\mathrm{t}+1) = \begin{cases} 1 & if\ rand < sig\left(V_{i,j}(\mathrm{t}+1)\right) \\ 0 & otherwise \end{cases} \tag{10}$$

where *rand* is a quasi-random number between (0, 1). This equation converts the representation of position form continuous to binary mode.

6 Chaotic sequences for inertia weight

The inertia weight (*w*) is the main element affecting the convergence to the best solutions. It also controls the balance between local search ability and the global exploration. The global search is facilitated with a large inertia weight, while with a small inertia weight, the local search is facilitated. The binary particle swarm optimization (BPSO) could be trapped in a local optimum which leads to results in premature convergence. To overcome this problem, chaotic map is used to change the value of inertia weight in each generation [27]. The logistic map is one of the most popular types of chaotic sequences. The logistic map equation is defined as following:

$$Z_{n+1} = \mu Z_n(1 - Z_n) \quad n = 1, 2, 3\ldots \tag{11}$$

where μ is the control parameter = 4; Z_1, Z_2, Z_3,…,Z_n are chaotic series generated by iteration and used to be the value of *w* in PSO [28].

7 Crowding distance computation

Crowding Distance Computation (CDC) is to estimate the density of the solutions surrounding a specific point in the generation. The CDC mechanism is used in this research to determine the deletion of non-dominated solutions from external archive when the archive is full. In addition, CDC maintains the diversity of non-dominated solutions in the external archive and also tries to balance between two objectives to select the optimal solutions. The CDC is estimated as following [29, 30]:

1. From the archive, obtain the number of non-dominated solutions
2. Initialize array of distance *D* for all particles (initialize with zero).
3. The solutions which have the highest and lowest fitness function values for each objective *EO* specified as infinite crowding distance values.

Set the D (1, EO) = D (end, EO) = infinity value.

4. Compute the CDC of each solution, the values of non-dominated solution in each objective are sorted in ascending order *SV*. Then, for each particle, the distance between the upper and the lower particles from this particle is calculated, as shown in Fig. 1. The CDC of particle *i* represents the sum of distances of particles *i* with add the sorted values $SV(i-1)$ and $SV(i+1)$ divided on the subtraction between the maximum and minimum value of each objective.
5. Sort the array of *D* in descending order. The particles which have the highest CDC will be chosen.

$$D(i) = D(i) + SV(i+1) - SV(i-1)/(\max - \min)$$

8 Crossover operation

Crossover is applied as a try to prevent falling in local optima. One-point crossover operation at which the crossover point is the middle of the solution is adopted in this research. This is the first time that the crossover is applied into the velocity and position of PSO.

9 Chaotic particle swarm optimization for discrete problem

For the purpose of investigating our hypothesis, Chaos Multi-Objective Discrete Particle Swarm Optimization (CMODPSO) is proposed and implemented. The following steps represent the proposed algorithm:

1. Initialize a random population of 100 particles (100 positions and 100 velocities). Each particle consists of the number of dimensions which are 50 dimensions and 5 types of maintenance (as shown in Table 1).
2. Convert the representation of positions form continuous to binary.
3. Initialize the local best position only before starting the generations. Knowing that the array of local best equals to the array of positions.
4. Evaluate the solutions using the objective function (fitness function) consisting of two-dimension arrays. The first one is for cost (using Eqs. 1, 2, 3) and the second one is for PCI value (using Eqs. 4 and 5).
5. Choose the best solutions by based on Pareto front as shown in (multi-objective optimization problems) and save the solutions in the external archive ***A***.
6. Check the external archive ***A*** if it is full or not. If ***A*** is full then the crowding distance computation (CDC) algorithm is used. In this paper, the capacity of archive is just 20 solutions.
7. Choose the global best position (the leader). In this paper, the sigmoid method is employed to choose the leader from the non-dominated solutions. The sigmoid method can be estimated as the following:

$$\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^1} \tag{12}$$

f_1, f_2 are fitness values for the first objective function and the second objective function respectively. The procedure of sigmoid method is given by the following:

   a. Calculate the $\sigma(j)$ for the members in ***A*** according to Eq. (12).
   b. Calculate $\sigma(i)$ for each particle using Euclidian distance (Dist) between $\sigma(1)$ and $\sigma(i)$. Then, calculate Euclidian distance between $\sigma(j)$ and $\sigma(i)$ (tempDist).
   c. Compare the (Dist) with all (tempDist), if tempDist $\leq$ Dist then Dist = tempDist. Otherwise, Dist will not change. The number of particles that have the lowest Dist represent the number of particles which are the leader [23].

8. Update the velocity of particles according to Eq. (6). To obtain the value of *w*, the logistic map as in Eq. (11) is used where the initial value of this equation $z(0)$ is equal to 0.7 [31].
9. Apply one-point crossover operation for the velocity. Then, a controlled mutation is used to keep the velocity in the range (− 6, 6) as this range is recommended by Kennedy and Eberhart for discrete problems [18].

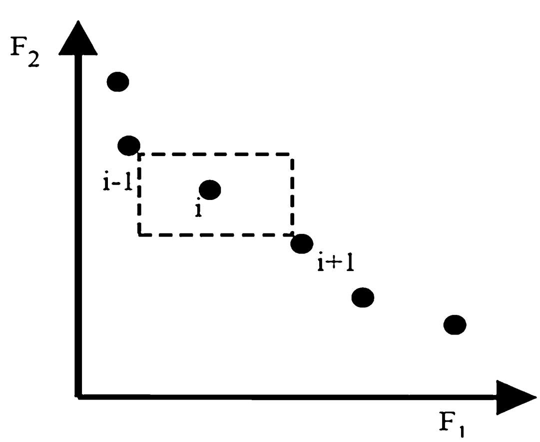


Fig. 1 Crowding distance computation

10. Update the position of particles using Eqs. (8, 9, 10).
11. Apply one-point crossover operation for positions. Then, mutation operator is applied to avoid falling in local optimal. The mutation is done by changing the position that has the value of 1, of some randomly selected particles.
12. Evaluate the fitness value of the new particle. The new solution is saved in order to compare it with the previous solution.
13. Update the local best solution using the following equation [32].

$$pbest_i(t+1) = \begin{cases} pbest_i(t), & \text{if } F(pbest_i(t)) < F(x_i(t+1)) \\ x_i(t+1), & \text{otherwise} \end{cases} \tag{13}$$

14. Pareto front is used again to choose the best solution. Then the new solutions and their positions are saved in the external archive.
15. After saving the new solutions, the external archive should be checked as in step 6.
16. Choose a new leader in the current iteration as in step 7.
17. Repeat steps 8–16 until the maximum number of iterations is reached. In this paper, the maximum number of iteration is equal to 100.

10 Compromise solution

Decision makers might make inaccurate decisions in real applications. Therefore, a fuzzy membership function is adopted to identify the compromise solution from the Pareto optimal set and also to improve the decision maker's preference. Considering a non-dominated solution Y_k in the archive, the satisfactory degree of Y_k for the ith objective function F_i is expressed by a membership function:

$$\mu_i^k = \begin{cases} 1, & F_i(Y_k) \le F_i^{\min} \\ \dfrac{F_i^{\max} - F_i(Y_k)}{F_i^{\max} - F_i^{\min}}, & F_i^{\min} < F_i(Y_k) < F_i^{\max} \\ 0, & F_i(Y_k) \ge F_i^{\max} \end{cases} \tag{14}$$

where F_i^{min} and F_i^{max} are the minimum and maximum of the ith objective function Fi. Then, the normalized membership function μ^k of Y_k is calculated by:

$$\mu^k = \frac{\sum_{i=1}^{M} \mu_i^k}{\sum_{k=1}^{|A|} \sum_{i=1}^{M} \mu_i^k} \tag{15}$$

Where M is the number of objective, |A| is the element number of the archive. The compromise solution is the one having the maximum of μ^k in the archive A [32].

11 Results

The proposed chaos multi-objective discrete particle swarm optimization (CMODPSO) is implemented to find the optimal maintenance plan for five pavement sections over 10 years. The results as shown in Fig. 2 shows that twenty non-dominated solutions are found after 100 generations. To simulate the agency preferences, the compromise solution which have the maximum membership value (μ) in the archive is selected as the optimal pavement maintenance. Table 2 shows the optimal maintenance of compromise solution after 100 generations. The maintenance cost of compromise solution is 183.58 while the sum of all residual PCI values is 3.1403e+09. The execution time is about two minutes, which is very short time compared with that achieved before that the process took 34.5 h [10] as show in Fig. 3. The results show significant improvement in the maintenance cost and pavement performance compared to the results obtained in the previous work that applied for the same problem [7] (399.25 for maintenance cost and 5.87e+10 for sum of all residual PCI values). The convergence of the developed algorithm to optimum solution could be achieved after 50 generations. Figures 4, 5, 6 and 7 shows the convergence pattern with number of generations.

To study effect of crossover operation on the performance of CMODPSO algorithm, the comparison between the non-dominated solutions found by the algorithm with and without using crossover operation is conducted. Figure 8 shows that some non-dominated solutions found by the algorithm without crossover operation stay without update during the iterations (between 80 and 100 generations). This means that it could be fallen in the local optima, while this case is not found when the crossover is considered as shown in Fig. 8. Table 3 shows the

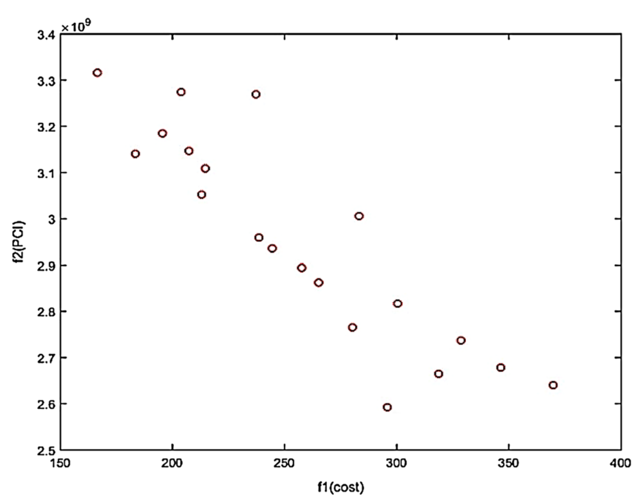


Fig. 2 The non-dominated solutions after 100 generations

Table 2 Optimal maintenance of compromise solution

Year 1					Year 2					Year 3					Year 4					Year 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
0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	0	1	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Year 6					Year 7					Year 8					Year 9					Year 10				
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
0	0	1	0	1	0	1	0	1	0	0	0	1	0	1	1	0	1	0	0	0	0	1	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

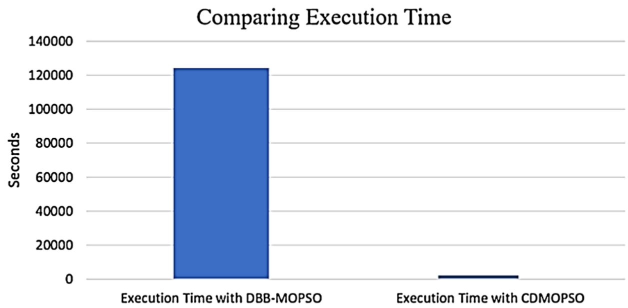


Fig. 3 The execution time of two algorithms

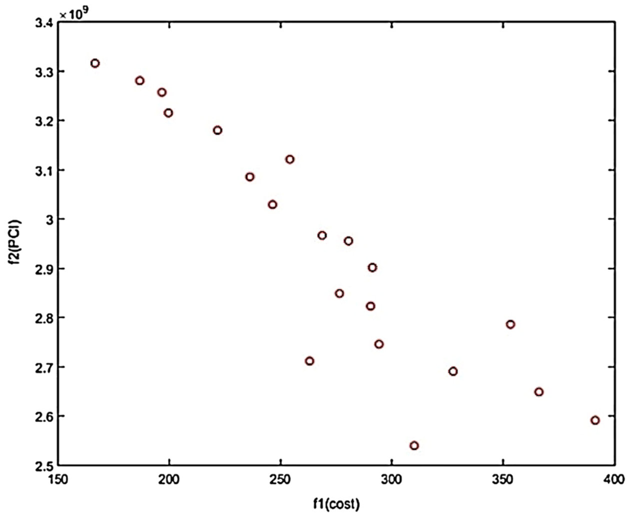


Fig. 4 The non-dominated solutions at 20 generations

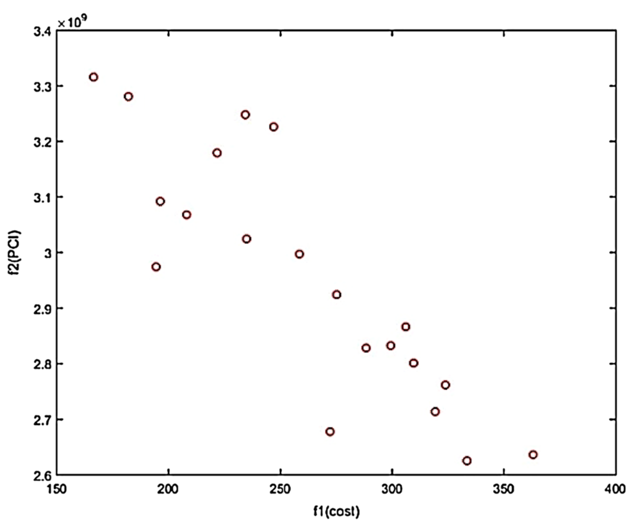


Fig. 5 The non-dominated solutions at 40 generations

comparison among twenty non-dominated solutions obtained at 100 generations with and without crossover operator. Based on Table 3, there is insignificant improvement in pavement performance objective for the most non-dominated solutions when the crossover operator is applied. However, there is noteworthy improvement in cost objective for the most non-dominated solutions found by considering crossover operator.

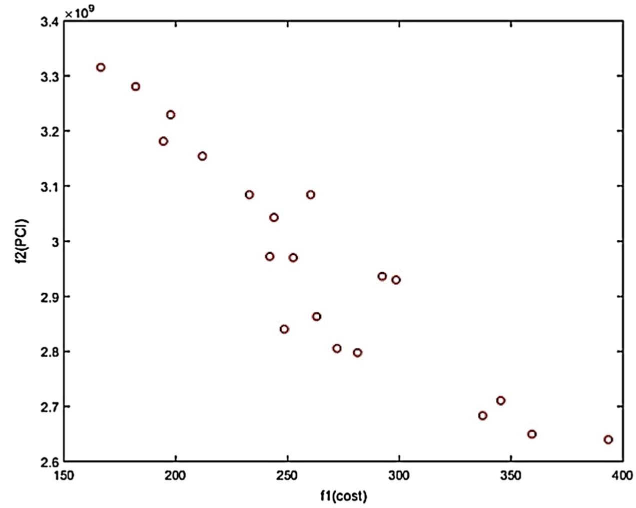


Fig. 6 The non-dominated solutions at 60 generations

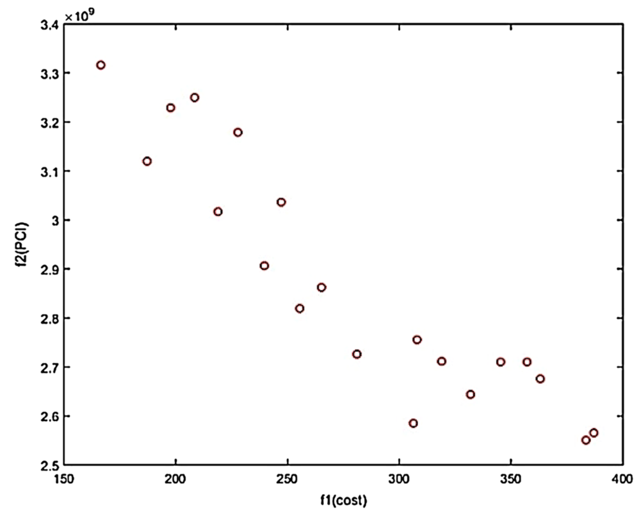


Fig. 7 The non-dominated solutions at 80 generations

12 Conclusions and future works

A novel CDPSO algorithm is developed by combining the crossover operation with velocity and position with multi objective PSO algorithm. The proposed algorithm is implemented to find the optimal pavement maintenance and rehabilitation scheduling considering two objectives: the minimization of the sum of all residual PCI values and the minimization of the total pavement rehabilitation cost.

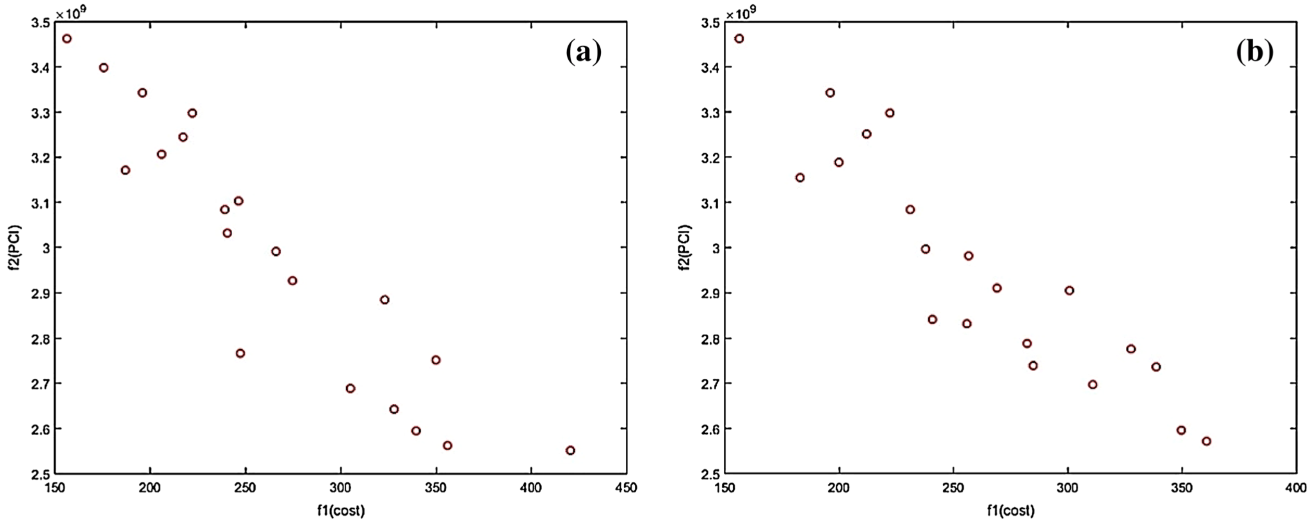


Fig. 8 **a** Show the solutions in generation 80, **b** show the solutions in generation 100

Table 3 Comparison between Non-dominated solutions with and without crossover operator

Non-dominated solution number	Solution in 100 generations with crossover		Solution in 100 generations without crossover	
	Cost	Sum of PCI	Cost	Sum of PCI
1	166.52	3.3157e+09	156.33	3.4631e+09
2	295.81	2.5927e+09	420.41	2.5517e+09
3	346.48	2.6783e+09	356.09	2.5619e+09
4	369.86	2.6408e+09	266	2.9915e+09
5	195.48	3.1852e+09	175.68	3.3983e+09
6	328.7	2.7373e+09	274.61	2.9271e+09
7	237.12	3.2697e+09	187.06	‘3.1711e+09
8	265.2	2.8619e+09	305.13	2.6892e+09
9	280.16	2.7656e+09	196.07	3.3426e+09
10	318.7	2.6651e+09	217.35	3.2441e+09
11	183.58	3.1403e+09	206.07	3.2061e+09
12	214.53	3.1083e+09	327.61	2.7755e+09
13	283.09	3.0053e+09	323.06	2.8849e+09
14	257.65	2.8946e+09	339.49	2.5953e+09
15	212.95	3.0521e+09	222.16	3.2973e+09
16	238.7	2.9595e+09	239.08	3.0833e+09
17	244.44	2.9354e+09	247.24	2.7669e+09
18	203.77	3.2738e+09	240.52	3.0323e+09
19	207.52	3.1467e+09	246.39	3.1035e+09
20	300.49	2.8165e+09	349.68	2.7516e+09

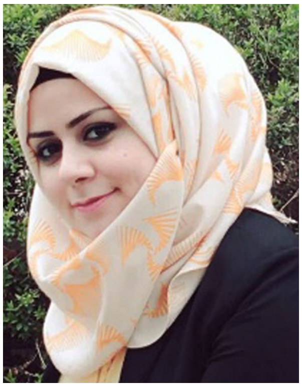
The developed algorithm has ability to address complex combinatorial optimization problem efficiently and find optimal solutions quickly. Therefore, the execution-time of the developed algorithm is too short about two minutes showing significant improvements in algorithm performance.

This is the first attempt to combine the crossover operation with CDPSO algorithm. The using of crossover operation in both velocity and position improves the performance of the algorithm and then avoid it to fall in local optima. Furthermore, using the ergodicity characteristics of chaotic variables instead of fixed value makes the value of inertia weight w in the best mode. This is further improvement in the developed algorithm performance.

For future work, for testing algorithm performance, the algorithm will be applied to the pavement maintenance scheduling problem along with another evolutionary algorithm. Moreover, the CDPSO algorithm will be implemented on constrained problems to examine the effects of using another chaos types on the solutions and the execution time.

References

1. Chang, J.R.: Particle swarm optimization method for optimal prioritization of pavement sections for maintenance and rehabilitation activities. Appl. Mech. Mater. **343**, 43–49 (2013)
2. Chen, C., Flintsch, G.W., Al-Qadi, I.L.: Fuzzy logic-based life-cycle costs analysis model for pavement and asset management. In: 6th Int. Conf. Manag. Pavements (2004)
3. Mohammed, M.A., Abd Ghani, M.K., Hamed, R.I., Mostafa, S.A., Ahmad, M.S., Ibrahim, D.A.: Solving vehicle routing problem by using improved genetic algorithm for optimal solution. J. Comput. Sci. **21**, 255–262 (2017)
4. Mohammed, M.A., et al.: Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution. J. Comput. Sci. **21**, 232–240 (2017)
5. Reséndiz, E., Rull-Flores, C.A.: Mahalanobis–Taguchi system applied to variable selection in automotive pedals components using Gompertz binary particle swarm optimization. Expert Syst. Appl. **40**(7), 2361–2365 (2013)
6. Tayebi, N.R., Moghadasnejhad, F., Hassani, A.: Analysis of pavement management activities programming by particle swarm optimization. In: Int. Conf. Adv. Electr. Electron. ACEEE, Trivandrum, Kerala, India **1643**(1), 149–154 (2010)
7. Elhadidy, A.A., Elbeltagi, E.E., Ammar, M.A.: Optimum analysis of pavement maintenance using multi-objective genetic algorithms. HBRC J. **11**(1), 107–113 (2015)
8. Chou, J.S., Le, T.S.: Reliability-based performance simulation for optimized pavement maintenance. Reliab. Eng. Syst. Saf. **96**(10), 1402–1410 (2011)
9. Mahmood, M.S.: Network—level maintenance decisions for flexible pavement using a soft computing-based framework, PhD Thesis. Nottingham Trent Univ. (2015)
10. Shen, Y., Bu, Y., Yuan, M.: "A novel chaos particle swarm optimization (PSO) and its application in pavement maintance decision. Ind. Electron. Appl. ICIEA **2009**, 3521–3526 (2009)
11. Moreira, A.V., Oliveira, J.R.M., Costa, L., Fwa, T.F.: Assessment of different genetic algorithms for pavement management systems. In: Proceedings of the Eighth International Conference on Maintenance and Rehabilitation of Pavements (2016)
12. Santos, J., Ferreira, A., Flintsch, G.: An adaptive hybrid genetic algorithm for pavement management. Int. J. Pavement Eng. **8436**(March), 1–21 (2017)
13. Fwa, T.F., et al.: Genetic-algorithm programing of road maintenance and rehabilitation. J. Transp. Eng. **3**, 246–253 (1996)
14. Rentz, H.H.; Life-cycle cost analysis in pavement design. Fed. Highw. Adm., p. 107 (1998)
15. Mahmood, M.S., Mathavan, S., Rahman, M.M.: Pavement maintenance decision optimization using a novel discrete bare-bones particle swarm algorithm. Nottingham Trent University, Nottingham (2016)
16. Fwa, T.F., Chan, W.T., Tan, C.Y.: Genetic-algorithm programing of road maintenance and rehabilitation. J. Transp. Eng. **122**, 246–253 (1996)
17. Shahin, M.Y., Walther, J.A.: Pavement maintenance management PAVER system, p. 278 (1990)
18. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. IEEE Int. Conf. **4**, 1942–1948 (1995)
19. Coello, C.C., Reyes-Sierra, M.: Multi-objective particle swarm optimizers: a survey of the state-of-the-art. Int. J. Comput. Intell. Res. **2**(3), 287–308 (2006)
20. Teodorović, D.: Swarm intelligence systems for transportation engineering: principles and applications. Transp. Res. Part C **16**(6), 651–667 (2008)
21. Reddy, M.J., Kumar, D.N.: Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. Hydrol. Process **2274**, 2897–2909 (2007)
22. De Carvalho, A.B., Pozo, A.: Measuring the convergence and diversity of CDAS multi-objective particle swarm optimization algorithms: a study of many-objective problems. Neurocomputing **75**(1), 43–51 (2012)
23. Mostaghim, S., Teich, J.: Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). IEEE Swarm Intell. Symp. **2**(5), 26–33 (2003)
24. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: 1997 IEEE Int. Conf. Syst. Man, Cybern. Comput. Cybern. Simul. **5**, 4104–4108 (1997)
25. Khanesar, M.A., Teshnehlab, M., Shoorehdeli, M.A.: A novel binary particle swarm optimization. In: Proc. 15th Mediterr. Conf. Control Autom. **1**(1), 3–8 (2007)
26. Xia, W., Jin, X., Dou, F.: Beam performance optimization of multibeam imaging sonar based on the hybrid algorithm of binary particle swarm optimization and convex optimization. Int. J. Antennas Propag. (2016)
27. Yang, C.-S., Chuang, L.-Y., Li, J.-C., Yang, C.-H.: Chaotic maps in binary particle swarm optimization for feature selection. In: 2008 IEEE Conf. Soft Comput. Ind. Appl., pp. 107–112 (2008)
28. Tang, X., Zhuang, L., Cai, J., Li, C.: Multi-fault classification based on support vector machine trained by chaos particle swarm optimization. Knowle.-Based Syst. **23**(5), 486–490 (2010)
29. Raquel, C.R., Naval, P.C.: An effective use of crowding distance in multiobjective particle swarm optimization. In: Proceeding GECCO'05 Proc. 7th Annu. Conf. Genet. Evol. Comput., pp. 257–264 (2005)
30. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
31. Zhou, Q., Zhang, W., Cash, S., Olatunbosun, O., Xu, H., Lu, G.: Intelligent sizing of a series hybrid electric power-train system based on chaos-enhanced accelerated particle swarm optimization. Appl. Energy **189**, 588–601 (2017)
32. Zhang, Y., Gong, D.-W., Ding, Z.: A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. Inf. Sci. **192**, 213–227 (2012)



Kawther Ahmed received the B.Sc. degree in Computer Science from University of Anbar, Ramadi, Iraq, in 2014. She is currently a master student in Computer Science Department, College of Computer Science and Information Technology, University of Anbar.

Belal Al-Khateeb received the B.Sc. (honors) (first class) degree in computer science from Al-Nahrain University, Baghdad, IRAQ, in 2000, and the M.Sc. degree in computer science from Al-Nahrain University, Baghdad, IRAQ, in 2003, and the Ph.D. degree from the School of Computer Science, University of Nottingham, Nottingham, U.K., in 2011. He is currently an Assist. Prof. at the College of Computer Science and Information Technology, University of Anbar. He has published over 37 refereed journal and conference papers. His current research interests include evolutionary and adaptive learning particularly in computer games, expert systems, and heuristics and meta/hyper-heuristics. He has a particular interest in computer games programming. Dr. Al-Khateeb is a reviewer of nine international journals (including two IEEE Transactions) and twelve conferences.

Maher Mahmood was born in Ramadi city, Anbar, Iraq, in 1978. He received the B.Sc. degree in civil engineering from the University of Anbar, Iraq, in 2000, M.Sc. degree in civil engineering from the University of Anbar, Iraq, in 2003 and the Ph.D. degree in highway engineering from Nottingham Trent University, Nottingham, UK, in 2015. From 2003 to 2015, he was an Assistant Lecturer with the Civil Engineering Department, University of Anbar. Since 2016, he has been a Lecturer with the Civil Engineering Department, University of Anbar, Iraq. He has carried out research in transportation and highway engineering. His research interests include transportation planning, highway engineering, pavement management and applied artificial intelligence.