J. Basic. Appl. Sci. Res., 2(7)6995-7002, 2012
© 2012, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A Multi-Objective Hybrid Optimization Algorithm for Project Selection Problem

Babak Amiri

Islamic Azad University, Fars Science and Research Branch

ABSTRACT

Choosing a portfolio of projects that meets an organization's objectives without exceeding available capital resources is recognized as a critical issue for which the decision maker takes several aspects into consideration. As most of these aspects may be conflicting, the problem can be considered as a multi-objective one. Correspondingly, in this study a special Multi-Objective Evolutionary Algorithm (MOEA) based on harmony search algorithm (HAS) is designed to solve the project selection problem. The original HAS often converges to local optima which is a disadvantage with this method. To avoid this shortcoming the HAS was combined with a Chaotic Local Search (CLS). In the proposed algorithm an external repository considered to save non-dominated solutions found during the search process and a fuzzy clustering technique was used to control the size of the repository. The experiment results show the capability of the proposed multi-objective algorithm in project selection problem.

1. INTRODUCTION

Selection of right sets of projects is considerably critical for organizations to successfully achieve their competitive advantages and corporate strategies. Due to limited resources and dynamic changes in business environment, this kind of selection is quite challenging for organizations. Beside one hundred selection tools and techniques, academics and practitioners have studied and recommended complex selection methodologies to facilitate the selection of right projects. Some of these methodologies have limitations in various aspects. Among the available useful approaches, optimization techniques (i.e. goal programming, multi-criteria decision-making, etc.) are the most fundamental quantitative tools for project portfolio selection which can address a high percentage of desired aspects. Doerner et al. [1] proposed a multi-objective Pareto Ant Colony Optimization which introduces Pareto Ant Colony Optimization as an especially effective meta-heuristic for solving the portfolio selection problem and considers the model introduced by Stummer and Heidenberger [2]. Badri, Davis, and Davis [3] introduced a comprehensive mixed 0–1 goal programming model for project selection in health service institutions. Mukherjee and Bera [4] provided an application of goal programming project selection decision with a case study from the Indian coal mining industry. They instituted a framework for incorporating ratings from experts to compute goal weights with normalization of deviational variables and stochastic demands. Santhanam and Kyparisis [5], proposed a multiple criteria decision model for information system project selection using a nonlinear 0–1 goal programming model that took advantage of hardware and software sharing for IS applications. Santhanam and Kyparisis [6], also discussed a nonlinear 0–1 decision model for interdependent information system project selection formulating benefit, resource, and technical interdependencies among candidate projects. Dey [7] proposed a project evaluation and selection by developing a decision support system. This DSS – applied in an Indian oil pipeline project-analysed project with respect to market, technicalities, and social and environmental impact in an integrated framework using analytic hierarchy process and a multiple-attribute decision-making technique. Gabriel, Kumar, Ordóñez, and Nasserian [8] proposed a unique multi-objective project selection model with probability distributions to describe costs and incorporating Monte Carlo simulation and AHP. Medaglia, Graves, and Ringuest [9] described an evolutionary approach for project selection problems with partially funded projects, multiple (stochastic) objectives, interdependencies in the objectives, and a linear structure for resource constraints. Mavrotas, Diakoulaki, and Kourentzis [10] proposed a two-phase method: (1) projects are ranked by a multi-criteria approach. (2) The pre-order of projects is used in an integer programming model to derive the final selection. Carlsson, Fullér, Heikkilä, and Majlender [11] presented a fuzzy mixed-integer programming model for R&D portfolio selection problem. Huang [12] incorporated random fuzzy uncertainty into project selection by integrating genetic algorithm with random fuzzy simulation.

According to the literature review, meta-heuristic approaches in solving project selection problems have a diminutive role in attracting researchers' attention. In this study, our contribution is to propose a hybrid multi-objective algorithm based on HSA and CLS. The harmony search algorithm was developed by Geem et al [13] and has successfully been applied to various combinatorial optimization problems [14-16]. It applies the musical process of searching for a perfect state of harmony. Musical harmony is analogous to the optimization solution

[*]**Corresponding author:** Babak Amiri, Islamic Azad University, Fars Science and Research Branch

vector, and the musician's improvisations are analogous to local and global search schemes in optimization techniques. Original HAS often converges to local optima, which is a disadvantage with this method. To avoid this shortcoming the HAS was combined with a Chaotic Local Search (CLS) in this paper. The reminder of this paper is organized as follows: fundamental definitions of multi-objective optimization are presented in Section 2. In Section 3, a description of project selection model is provided. Section 4 deals with the proposed multi-objective algorithm. The computational results are provided in Section 5. Finally, a summary of conclusions is explained in Section 6.

2. Multi-objective optimization

In a multi-objective optimisation problem, the purpose is to optimise several conflicting objectives simultaneously while still meeting some constraints. The project selection problem algorithm can be formulised as a multi-objective problem. The multi-objective problem can be described as follows [17][18]:

$\min F = [f_1(X), f_2(X), \dots, f_n(X)]$

where $f_i(X)$ is the ith objective function and X is the vector of the optimization variables, n is the number of objective functions.

The solution to the multi-objective optimisation problem is a set of Pareto points. In the multi-objective optimisation problem, a solution $X^* \in \Omega$ is a Pareto optimal if there is no solution (X) in Ω such that X dominates X*. Ω is the set of all feasible values of X. The solution X1 is said to dominate the solution X2 if

$\forall\, j \in \{1, 2, \dots, n\}, f_j(X_1) \le f_j(X_2)$

$\exists\, k \in \{1, 2, \dots, n\}, f_k(X_1) \le f_k(X_2)$

Solutions which dominate others but not themselves, are called non-dominated solutions.

3. Formulation of the project selection problem

In this paper *N* projects with interdependencies are considered for selection. Three objectives have been considered for project selection: maximizing the total benefit, minimizing the total cost, and minimizing the total risk from the selected projects. That these objectives are contradictory is obvious (To increase benefit, one should attempt to decrease cost and risk). Interdependencies are considered in the benefit and cost objectives while calculating risk interdependencies is rather difficult and not very meaningful since risk is derived as an average of several dimensions. We assumed that implementing interdependent projects will cause a positive synergy (an additional benefit apart from what is gained from each project individually or a reduction in costs).

3.1. Parameters

N Number of projects, $i, j = 1, \dots, N$
T Number of periods, $t = 1, \dots, T$
b_{it} The derived benefit from implementing project *i* alone in period *t*
b_{ij} The additional benefit from implementing projects *i* and *j* together
c_{it} The required cost by project *i* in period *t*
c_{ij} The shared cost by projects *i* and *j*
r_{it} The risk of implementing project *i* in period *t*
S_m The set of mandatory projects
S_e The e^{th} set of exclusive projects
A_i The set of projects that the implementing of project *i* is contingent upon the implementation of all of them
Q_i The set of projects that the implementing of project *i* is contingent upon the implementation of at least one of them

3.2. Decision variable

$$X_{it} \begin{cases} 1, & if\ project\ i\ is\ selected\ in\ period\ t \\ 0, & therwise \end{cases}$$

3.3. Mathematical formulations

Maximizing the total benefit

$$Maximize \quad \sum_{i=1}^{N}\sum_{t=1}^{T} b_{it}X_{it} + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} b_{ij}\sum_{t=1}^{T} X_{it}\cdot\sum_{t=1}^{T} X_{jt} \qquad (1)$$

Minimizing the total cost

$$\text{Minimize} \quad \sum_{i=1}^{N}\sum_{t=1}^{T} c_{it}X_{it} + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}\sum_{t=1}^{T} X_{it}\cdot\sum_{t=1}^{T} X_{jt} \qquad (2)$$

Minimizing the total risk

$$\text{Minimize} \quad \sum_{i=1}^{N}\sum_{t=1}^{T} r_{it}X_{it} \qquad (3)$$

S.t:

$$\sum_{t=1}^{T} X_{it} = 1 \qquad \forall i \in S_m \tag{4}$$

$$\sum_{t=1}^{T} X_{it} \leq 1 \qquad \forall i \tag{5}$$

$$\sum_{i \in S_e} \sum_{t=1}^{T} \leq 1 \qquad \forall e = 1, \dots, E \tag{6}$$

$$|A_i| \sum_{t=1}^{T} X_{it} \leq \sum_{j \in A_i} \sum_{t=1}^{T} X_{jt} \tag{7}$$

$$\sum_{t=1}^{T} X_{it} \leq \sum_{j \in Q_i} \sum_{t=1}^{T} X_{jt} \tag{8}$$

The objective function given in Eq. (1) represents the total benefit derived from the implemented projects. The first term is the total benefit derived from each implemented project individually. The second term is an additional benefit derived from implementing projects *i* and *j* together. The Eq. (2) represents the total cost objective function. The first term is the total cost required by each implemented project individually. The second term is the cost shared by implementing projects *i* and *j* together. The objective function given in Eq. (3) represents the total risk of the selected projects. The Eq. (4) ensures that each mandated project is done. Constraint (5) guarantees that each project will not be implemented more than one. Constraint (6) specifies that at most one project from every exclusive set is selected. Eq. (7) indicates the situation in which execution of a given project *i* is contingent upon the implementation of all of the projects in a specific set (A_i) where $|A_i|$ is the number of elements in the set A_i. Eq. (8) defines a condition where implementation of a given project *i* is contingent upon implementation of at least one of the projects in a specific set (Q_i).

4. Multi-objective Harmony Search Algorithm

4.1. Original Harmony search algorithm

The harmony search algorithm is a new algorithm that can be conceptualised from a musical performance (say, a jazz improvisation) that involves searching for a better harmony. Just like music improvisation seeks a best state (fantastic harmony) determined by aesthetic estimation, the improvisation process seeks the best state (global optimum) determined by an objective evaluation of the function. As an aesthetic estimation is determined by the set of pitches played by ensemble instruments, evaluating the function was determined by the set of values assigned to the decision variables, just as the aesthetic sound quality can be improved by constant practice, the value of the objective function can be improved iteration by iteration [13].

Each musician (saxophonist, double bassist, and guitarist) can correspond to each decision variable (x1, x2, and x3), and the range of each musical instrument (saxophone={Do, Re, Mi}; double bass={Mi, Fa, Sol}; and guitar={Sol, La, Si}) corresponds to the range of each variable value (x1={100, 200, 300}; x2={300, 400, 500}; and x3= {500, 600, 700}). If the saxophonist toots the note Do, the double bassist plucks Mi, and the guitarist plucks Sol, their notes together make a new harmony (Do, Mi, Sol). If this New Harmony is better than the existing harmony, the New Harmony is kept. Likewise, the new solution vector (100, 300, 500) generated in the improvisation process is kept if it is better than the existing harmony in terms of the objective function value. Just as the quality of the harmony is enhanced practice after practice, the quality of the solution is enhanced iteration by iteration.

According to the above algorithmic concept and the procedure for searching for harmony consists of the following steps [18]:

Step 1. Initialise the problem and algorithm parameters.
Step 2. Initialise the harmony memory.
Step 3. Improvise a new harmony.
Step 4. Update the harmony memory.
Step 5. Check the stopping criterion.

In the next section a description of the proposed multi-objective algorithm is given.

4.2. Proposed hybrid algorithm

Previously in this work, the search capability of the HSA algorithm and CLS was specifically used to solve the project selection problem. The steps of the proposed hybrid algorithm (HSA-CLS) are as follows:

Step 1: Initializing the problem and algorithm parameters

The HS algorithm parameters are also specified in this step. These are the harmony memory size (HMS), or the number of solution vectors in the harmony memory, the harmony memory considering rate (HMCR), the pitch adjusting rate (PAR), and the number of improvisations (NI), or stopping criterion. The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored.

Step 2: Initialise the harmony memory

In this step the HM matrix is filled with as many randomly generated solution vectors as the HMS, by considering the range of values for each variable. Considering the possible range of values to improve the convergence of the method because the space of the possible solutions is restricted, we have:

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^1 & x_N^1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix} \tag{9}$$

Step 3: Apply the CLS on the HM

In order to make the algorithm perform better and to increase the ability to search, we incorporated the chaotic local search (CLS) into HSA. Chaos is characterised as ergodicity, randomicity and regularity. Because chaos queues can experience all the states in a specific area without repeat, chaotic search becomes a novel tool used as an optimiser [19].

The tent equation is a well known chaos system that can also be introduced to the chaotic local search process, which could be defined by the following equation:

$$\begin{aligned} &Cx_i = [cx_i^1, cx_i^2, \dots, cx_i^n]_{1\times n}, \quad i = 0, 1, 2, \dots, N_{chaos} \\ &cx_{i+1}^j = \begin{cases} 2cx_i^j, & 0 < cx_i^j < 0.5 \\ 2(1 - cx_i^j) & 0.5 < cx_i^j < 1 \end{cases} \quad j = 1, 2, \dots, n \\ &cx_i^j \in [0,1], cx_0^j \notin \{0.25, 0.5, 0.75\} \\ &cx_0^j = rand(.) \end{aligned} \tag{10}$$

where cx_i^j indicates the j^{th} chaotic variable, N_{choas} is the number of individuals for CLS and *rand* (·) is a random number between [0,1].

At first, one of non-dominated solutions in the *HM* was considered as an initial population for CLS (X_{cls}^0) randomly. X_{cls}^0 was scaled into [0,1] according the following equation:

$$\begin{aligned} &X_{cls}^0 = [x_{cls,0}^1, x_{cls,0}^2, \dots, x_{cls,0}^n]_{1\times n} \\ &Cx_0 = [cx_0^1, cx_0^2, \dots, cx_0^n] \\ &cx_0^j = \frac{x_{cls,0}^j - x_{min}^j}{x_{max}^j - x_{min}^j}, \quad j = 1, 2, \dots, n \end{aligned} \tag{11}$$

where x_{min}^j and x_{max}^j are the minimum and maximum values of state variables vector.
Then, the chaos population for CLS is generated as follows:

$$\begin{aligned} &X_{cls}^i = [x_{cls,i}^1, x_{cls,i}^2, \dots, x_{cls,i}^n]_{1\times n}, \quad i = 1, 2, \dots, N_{chaos} \\ &x_{cls,i}^j = cx_{i-1}^j \times (x_{max}^j - x_{min}^j) + x_{min}^j, \quad j = 1, 2, \dots, n \end{aligned} \tag{12}$$

The objective functions for all the individuals of CLS were evaluated, and the non-dominated solution in CLS was selected and replaced with a dominated solution in the *HM.*

Step 4: Store the non-dominated solutions into a repository and use fuzzy clustering to control the size of repository

A fuzzy-based clustering procedure was normalized to control the size of the repository. In the procedure, a fuzzy membership function was used to normalize the best compromise solution. In other words, decision making was done when the repository was filled. For any individual in the repository, the membership function of each objective function is defined as follows:

$$\mu_{fi}(\boldsymbol{X}) = \begin{cases} 1 & for\ f_i(\boldsymbol{X}) \le f_i^{min} \\ 0 & for\ f_i(\boldsymbol{X}) \ge f_i^{max} \\ \dfrac{f_i^{max} - f_i(\boldsymbol{X})}{f_i^{max} - f_i^{min}} & f_i^{min} \le f_i(\boldsymbol{X}) \le f_i^{max} \end{cases} \quad (13)$$

where f_i^{min} and f_i^{max} are the upper and lower limits of each objective function, respectively. In the proposed algorithm, the values of f^{min} and f^{max} were evaluated using the results achieved by normalization of each objective function separately. For each individual in the repository, the5normalized membership value was evaluated using

$$N_\mu(j) = \frac{\sum_{k=1}^{n} w_k \times \mu_{fk}(\boldsymbol{X}_j)}{\sum_{j=1}^{m}\sum_{k=1}^{n} w_k \times \mu_{fk}(\boldsymbol{X}_j)} \quad (14)$$

where m is the number of non-dominated solutions, n the number of objective functions, and w_k is the weight for the k^{th} objective function. This membership function shows a type of decision making criterion that is adaptive and changes with the available decision options.

Step 5: Improvise a new harmony

A new harmony vector $x' = (x_1', x_2', \dots, x_N')$ was generated based on three rules: (1) memory consideration, (2) pitch adjustment, and (3) random selection. Generating a new harmony is called 'improvisation' [12].

In the memory consideration, the value of the first decision variable x_1' for the new vector was chosen from any of the values in the specified *HM* range $(x_1^1 \dots x_1^{HMS})$ *HMS*. The values of the other decision variables were chosen in the same way. The *HMCR*, which varied between 0 and 1, was the rate of choosing one value from the historical values stored in the *HM*, while *(1-HMCR)* was the rate of randomly selecting one value from the possible range of values.

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & with\ probability\ \ HMCR \\ x_i^1 \in X_i & with\ probability\ \ (1 - HMCR) \end{cases} \quad (15)$$

For example, a *HMCR* of 0.85 indicates that the *HS* algorithm will choose the decision variable value from historically stored values in the *HM* with an 85% probability or from the entire possible range with a (100–85) % probability. Every component obtained by memory consideration was examined to determine whether it should be pitch-adjusted. This operation used the *PAR* parameter, which is the rate of pitch adjustment as follows:

$$Pitch\ adjusting\ decision\ for\ x_i' \leftarrow \begin{cases} Yes & with\ probability\ PAR \\ No & with\ probability (1 - PAR) \end{cases} \quad (16)$$

The value of *(1 – PAR)* sets the rate of doing nothing. If the pitch adjustment decision for x_i' is YES, x_i' is replaced as follow:

$$x_i' \leftarrow x_i' \pm rand() \times bw \quad (17)$$

where *bw* is an arbitrary distance bandwidth, *rand()* is a random number between 0 and 1.

Step 6: Calculate objective functions

In this step the new solution should be decoded to identify all the components of the corresponding graph and then the objective functions will be calculated.

Step 7: Is the new harmony Pareto dominates?

If the new solution is Pareto dominated, go to the next step, otherwise go to step 9.

Step 8: Update harmony memory

If the new harmony vector $x' = (x_1', x_2', \dots, x_N')$ dominates one of the solutions in the *HM* then swap the new harmony with the dominated solution in the *HM* and add the solution to the repository.

Step 9: Check stopping criterion

If the stopping criterion (maximum number of improvisations) is satisfied, computation is terminated. Otherwise, go to step 5.

5. EXPERIMENTAL RESULTS

The performance of the proposed multi-objective HSA-CLS is compared with a well-known multi-objective genetic algorithm SPEA II on synthetic benchmark datasets. The effectiveness of stochastic algorithms is greatly dependent on the generation of initial solutions and therefore, for every dataset, algorithms have individually performed 100 times to test their own effectiveness, and each time with randomly generated initial solutions. Our algorithm was implemented into Matlab 7.1. All the experiments were conducted on a computer with Intel Core 2 Duo, 2.66 GHz, 4 GB RAM.

5.1. SPEA II

The Strength Pareto Evolutionary Algorithm II (SPEA II) (proposed by Zitzler et al. [20]) is an improved extension of SPEA which differs from it in three following aspects: fitness assignment technique, density estimation technique, and archive truncation method. This genetic algorithm is implemented according to the description in the literature. In genetic algorithms chromosomes exchange information with each other by crossover or/and mutation, yet according to the specific nature of our solution, application of these operations require a lot of modification after solutions have undergone crossover or/and mutation. This makes the process unpractical and clearly undermines the positive effect of crossover or/and mutation.

5.2. Test problems

Table 1 represents 10 test problems generated randomly. After carrying out the test problems, the performance of the proposed multi-objective HSA-CLS is measured using some comparison metrics which are discussed in the following section.

Table 1. Test problems

No. test problems	No. Projects	No. Periods	Benefit	Cost	Risk
1	15	3	U[0, 35]	U[0, 35]	U[0, 1]
2	15	1	U[0, 50]	U[0, 30]	U[0, 1]
3	20	3	U[0, 50]	U[0, 30]	U[0, 1]
4	20	1	U[0, 50]	U[0, 30]	U[0, 1]
5	25	3	U[0, 55]	U[0, 60]	U[0, 1]
6	25	2	U[0, 45]	U[0, 30]	U[0, 1]
7	25	1	U[0, 80]	U[0, 60]	U[0, 1]
8	30	3	U[0, 100]	U[0, 80]	U[0, 1]
9	30	2	U[0, 95]	U[0, 75]	U[0, 1]
10	30	1	U[0, 120]	U[0, 100]	U[0, 1]

5.4. Comparison metrics

1. Number of non-dominated solutions: This metric shows the number of locally non-dominated solutions that each algorithm can find. It’s obvious that a higher value of this metric signifies better exploration and a more diverse search direction.
2. Quality metrics: This metric is simply measured by putting together the non-dominated solutions found by the algorithms, and the ratios between non-dominated solutions are reported.
3. Diversity metric: This metric measures the spread of obtained solutions. As a result, the volume of the smallest cube containing all the non-dominated solutions can be considered as a criterion to evaluate this metric.
4. Spacing metric: We define this metric as follows:

$$S = \frac{\sum_{i=1}^{N-1}\left|d_i - \bar{d}\right|}{(N-1)\bar{d}} \tag{18}$$

where d_i is the Euclidean distance between the consecutive solutions in the obtained non-dominated set of solutions, and $\bar{d}$ is the average of these distances. This metric allows us to measure the uniformity of the solution set points’ spread. Using a triangularization technique or a Voronoi diagram approach to calculate d_i, the above procedure can be extended to estimate the spread of solutions in higher dimensions. To apply the triangularization technique in a three dimensional space, we divided the space by the Delaunay triangulation into some tetrahedrons, the vertices of which were the obtained non-dominated solutions. Note that the Delaunay triangulation is a set of lines connecting each point to its natural neighbours. Afterwards, we substituted the term d_i in the above formula by the volume of these tetrahedrons.

5.5. Computational results

Table 2 shows the number of non-dominated solutions gained by HSA-CLS and SPEA II. It can be observed that HSA-CLS outperforms SPEA II according to this metric, for it produces a higher number of non-dominated solutions. In addition, as shown in Table 3, the proposed HSA-CLS has generally achieved solutions with higher quality in comparison with SPEA II. In Table 4, the comparison of diversification metric indicates the dominance of HSA-CLS to SPEA II. Finally, Table 5, which represents the spacing metric, signifies a higher performance for SPEA II, while the difference between HSA-CLS and SPEA II in this metric is practically negligible. Tables 2–4 are represented in an average term, yet Table 5 shows the minimum of the spacing metric gained in each test problem. It is important to mention that the maximum of the spacing metric, in the test problems in which SPEA II outperformed HSA-CLS, was notably higher in comparison to that of HSA-CLS.

Due to the discrete nature of problem and large number of constraints, the trade-off surface has some holes and thus leads to difficulty in spacing metric interpretation. Finally, it is reasonable to claim the superiority of the designed HSA-CLS to SPEAII.

Table 2. Comparison of number of non-dominated solutions

Problem	HSA-CLS	SPEAII
1	32.7	25.2
2	35.3	25.6
3	38.9	33.8
4	29	25.3
5	69.5	42.7
6	48.1	45.7
7	91.3	51.1
8	72.9	58
9	86.7	58.1
10	145.1	58.6

Table 3. Comparison of number of non-dominated solutions

Problem	HSA-CLS:SPEAII
1	31.7:23.6
2	32.5:22.6
3	33.9:32.8
4	21.6:21.6
5	56.6:37.1
6	49.3:41.2
7	75.7:41.4
8	49.1:45.7
9	64.4:43.6
10	109.2:38.9

Table 4. Comparison of number of non-dominated solutions

Problem	HSA-CLS	SPEAII
1	1.9208×10^5	1.4357×10^5
2	3.4722×10^5	1.8441×10^5
3	5.1635×10^5	4.1145×10^5
4	7.3307×10^5	4.9857×10^5
5	1.6915×10^5	1.0137×10^5
6	1.1591×10^5	3.0792×10^5
7	1.1537×10^5	1.0083×10^5
8	7.6811×10^5	5.7617×10^5
9	2.7719×10^5	2.4194×10^5
10	7.1185×10^5	5.9359×10^5

Table 5 Comparison of number of non-dominated solutions

Problem	HSA-CLS	SPEAII
1	0.0015	0.0005
2	0.0173	0.0060
3	0.0182	0.0095
4	0.0241	0.0220
5	0.3364	0.1789
6	0.0287	0.0511
7	0.4290	0.3320
8	0.3366	1.8624
9	1.5779	0.4960
10	1.9055	0.9099

6. Conclusions

This paper presented a hybrid multi-objective harmony search algorithm to solve the project selection problems with simultaneously considering three objectives: maximizing the total benefit, minimizing the total cost, and minimizing the total risk. The original HAS often converges to local optima which is a disadvantage with this method. To avoid this shortcoming the HAS was combined with a Chaotic Local Search (CLS). In the proposed algorithm an external repository considered to save non-dominated solutions found during the search process and a fuzzy clustering technique was used to control the size of the repository. The experiment results show the capability of the proposed multi-objective algorithm in project selection problem.

REFERENCES

[1] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection," *Annals of Operations Research,* vol. 131, pp. 79-99, 2004.

[2] C. Stummer and K. Heidenberger, "Interactive R&D portfolio selection considering multiple objectives, project interdependencies, and time: A three-phase approach," 2001, pp. 423-428 vol. 2.

[3] M. A. Badri and D. Davis, "A comprehensive 0-1 goal programming model for project selection," *International Journal of Project Management,* vol. 19, pp. 243-252, 2001.

[4] K. Mukherjee and A. Bera, "Application of goal programming in project selection decision--A case study from the Indian coal mining industry," *European Journal of Operational Research,* vol. 82, pp. 18-25, 1995.

[5] R. Santhanam and G. J. Kyparisis, "A decision model for interdependent information system project selection," *European Journal of Operational Research,* vol. 89, pp. 380-399, 1996.

[6] R. Santhanam and J. Kyparisis, "A multiple criteria decision model for information system project selection," *Computers & Operations Research,* vol. 22, pp. 807-818, 1995.

[7] P. K. Dey, "Integrated project evaluation and selection using multiple-attribute decision-making technique," *International Journal of Production Economics,* vol. 103, pp. 90-103, 2006.

[8] S. A. Gabriel, S. Kumar, J. Ordóñez, and A. Nasserian, "A multiobjective optimization model for project selection with probabilistic considerations," *Socio-Economic Planning Sciences,* vol. 40, pp. 297-313, 2006.

[9] A. L. Medaglia, S. B. Graves, and J. L. Ringuest, "A multiobjective evolutionary approach for linearly constrained project selection under uncertainty," *European Journal of Operational Research,* vol. 179, pp. 869-894, 2007.

[10] G. Mavrotas, D. Diakoulaki, and A. Kourentzis, "Selection among ranked projects under segmentation, policy and logical constraints," *European Journal of Operational Research,* vol. 187, pp. 177-192, 2008.

[11] C. Carlsson, R. Fullér, M. Heikkila, and P. Majlender, "A fuzzy approach to R&D project portfolio selection," *International Journal of Approximate Reasoning,* vol. 44, pp. 93-105, 2007.

[12] X. Huang, "Optimal project selection with random fuzzy parameters," *International Journal of Production Economics,* vol. 106, pp. 513-522, 2007.

[13] Z. W. Geem and J. H. Kim, "A new heuristic optimization algorithm: harmony search," *Simulation,* vol. 76, pp. 60-68, 2001.

[14] Z. Geem, "Harmony search in water pump switching problem," *Advances in Natural Computation,* pp. 445-445, 2005.

[15] R. Forsati, A. Haghighat, and M. Mahdavi, "Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing," *Computer Communications,* vol. 31, pp. 2505-2519, 2008.

[16] M. Saka, "Optimum geometry design of geodesic domes using harmony search algorithm," *Advances in Structural Engineering,* vol. 10, pp. 595-606, 2007.

[17] C. M. Lin and M. Gen, "Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm," *Expert Systems with Applications,* vol. 34, pp. 2480-2490, 2008.

[18] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer methods in applied mechanics and engineering,* vol. 194, pp. 3902-3933, 2005.

[19] B. Li and W. Jiang, "Chaos Optimization Method and Its Application [J]," *Control Theory & Applications,* vol. 4, 1997.

[20] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," 2001.