

A comparative study of marriage in honey bees optimisation (MBO) algorithm in multi-reservoir system optimisation

Alireza B Dariane* and Zeinab Farahmandfar

Department of Civil Engineering, KN Toosi University of Technology, Tehran, Iran

ABSTRACT

Contemporary reservoir systems often require operators to meet a variety of goals and objectives; these in turn frequently complicate water management decision-making. In addition, many reservoir objectives have non-linear relationships and are therefore difficult to implement using traditional optimisation techniques. A practical application of the marriage in honey bees optimisation (MBO) algorithm is being utilised for Karkheh multi-reservoir system, south-western Iran, where supplying irrigation water for agricultural areas and maintaining a minimum in-stream flow for environmental purposes is desired. Optimal monthly reservoir release information by MBO is highlighted and the results compared to those of other evolutionary algorithms, such as the genetic algorithm (GA), ant colony optimisation for continuous domains (ACO_R), particle swarm optimisation (PSO) and elitist-mutation particle swarm optimisation (EMPSO). The results indicate the superiority of MBO over the algorithms tested.

Keywords: non-linear optimisation, multi reservoir system, honey bee mating optimisation algorithm, evolutionary algorithms, meta-heuristic algorithms

INTRODUCTION

In recent years, evolutionary algorithms have become widely popular in water resource system optimisation. Some of these algorithms concentrate on social animals, such as ant colony optimisation, particle swarm optimisation and bee-based algorithms. A group of social animals can be considered as a dynamic system that collects outside information and adjusts behavior in consideration of this. Each group of social animals accomplishes their specific tasks with regard to their specific biological properties. The honey bee is one of the most popular social animals considered in swarm intelligence studies. Among different honey bee activities, foraging, nest site selection and mating are the most important fields used to create artificial algorithms.

Yonezawa and Kikuchi (1996) examined the foraging behavior of honey bees and constructed an algorithm to indicate the importance of their group intelligence principle. This algorithm is simulated with 1 and 3 foraging bees and the results showed that 3 foraging bees were faster than 1 foraging bee at decision making processes. Sato and Hagiwara (1997) proposed an improved genetic algorithm based on foraging behavior of honey bees and named it the 'bee system'. In experimental studies they compared the Bee System with the conventional genetic algorithm and found that the bee system performed better, especially for highly complex multivariate functions. Lucic and Teodorovic (2001) published the first study in which the bee system was applied to the travelling salesman problem (TSP). The performance of the algorithm was tested on 10 benchmark problems. Experimental results showed that in all instances with less than 100 nodes the Bee System produced the optimal solution in a very short time.

Yang (2005) presented a virtual bee algorithm (VBA) based on foraging behavior of honey bees. The VBA algorithm was tested on 2 functions with 2 parameters and the results show that, although the proposed algorithm performs similarly to the genetic algorithm, it is much more efficient. Karaboga (2005) analysed the foraging behavior of honey bee swarms and proposed a new algorithm, called the 'artificial bee colony' (ABC), simulating this behavior for solving multi-dimensional and multi-modal benchmark optimisation problems. Basturk and Karaboga (2006) expanded the experimental results of Karaboga (2005) and tested the performance of the algorithm on 5 multi-dimensional benchmark functions. The results were compared with those of the other algorithms. It was pointed out that ABC algorithm outperformed the genetic algorithm for functions having both multi-modality and uni-modality.

Pham et al. (2006a) proposed the 'bees algorithm', inspired by the natural foraging behaviour of honey bees. The algorithm was applied to 8 benchmark functions and the results were compared with those of the deterministic simplex method, stochastic simulated annealing, genetic algorithm and the ant colony system. The bees algorithm generally outperformed other techniques in terms of solution speed and accuracy of results.

Lemmens (2006) then investigated whether the ant colony algorithm was outperformed by the bee colony algorithm. The results of the experiments showed that:

- the ant colony algorithms use less time per iteration step in small-sized worlds;
- the bee colony algorithms are significantly faster in finding and collecting foods and use fewer time steps to complete the task
- with growing group sizes, the bee colony algorithm eventually outperforms ant colony algorithms on a time per iteration step measure.

Seeley and Buhman (1999) investigated the nest site selection behavior of honey bee colonies for the first time. They repeated the observations of Lindauer (1955) by taking advantage of modern techniques. They observed that the nest site selection

* To whom all correspondence should be addressed.

+88770006; e-mail: borhani@kntu.ac.ir

Received 15 February 2012; accepted in revised form 27 February 2013.

process starts with sending several hundred scout bees in search of potential nest sites. The scouts then return to the cluster, report their findings by means of waggle dances, and the colony decides on the new nest site. The type of waggle dance depends on the quality of the site being advertised. Recently, Diowld et al. (2010) used nest site selection behavior in honey bees to construct an optimisation algorithm and showed that the algorithm has a high potential for solving optimisation problems.

Honey bee mating optimisation

Each bee in a colony does a sequences of tasks based on its genetic condition, ecological environment and social regulations (Rinderer and Collins, 1986). The queen is the most important member in the hive as she produces the new generation, being specialised in laying eggs. The queen mates with multiple drones for several days, once in her lifetime. The sperm of each drone is stored within the queen's body, in the spermatheca. She uses the stored sperm to produce both fertilised and unfertilised eggs. Drones are the fathers of the colony and emerge from unfertilised eggs. Their main task is to transfer sperm to the queen's spermatheca; soon after the mating process they die. Workers mainly take care of new broods but sometimes also lay eggs as they emerge from unfertilised eggs. The mating process occurs during a mating flight, far from the nest. In a typical mating, the queen mates with 7 to 20 drones. Each time the queen lays a fertilised egg, she receives a random mixture of the drones' sperm from her spermatheca (Abbass, 2001a).

The artificial algorithm starts with generating broods at random and, because the queen is always a 'better' bee, the best of the broods is selected as the queen. After queen selection, a set of mating flights begin. The mating flight can be visualised as a set of transitions in a state-space (the environment), where the queen moves between different states in a space with some speed and mates with the drone encountered at each state. This probabilistic process is modeled using the following equation:

$$\text{Prob}(Q, D) = e^{-\frac{\Delta(f)}{S(t)}} \quad (1)$$

where:

- prob(Q,D) is the probability of adding the sperm of Drone D to the spermatheca of Queen Q; that is, the probability of a successful mating
- $\Delta(f)$ is the absolute difference between the fitness of D (i.e., $f(D)$) and the fitness of Q (i.e., $f(Q)$)
- $S(t)$ is the speed of the queen at time t .

This function acts as an annealing function, where the probability of mating is high when the queen is still at the beginning of her mating flight and her speed is therefore high, or when the fitness of the drone is as high as that of the queen. However, the probability function acts only when the queen is fitter than the drone; if the drone is fitter than the queen his sperm is directly transmitted to the queen's spermatheca (Teo and Abbass, 2001).

The queen has a specific spermatheca size and initial energy and speed. The energy and speed of the queen are initialised at random at [0.5, 1] at the start of each mating flight, and the queen returns to her nest when the energy is within some threshold from zero or when her spermatheca are full. After each mating flight, the queen's speed and energy decreases and this is shown using the following equations (Abbass, 2001a).

$$\text{Speed}(t + 1) = \alpha \times \text{Speed}(t) \quad (2)$$

$$\text{Energy}(t + 1) = \text{Energy}(t) - \gamma \quad (3)$$

where:

- α is a factor between [0, 1]
- γ is the amount of energy reduction after each transition and is computed using the following equation (Abbass, 2001a)

$$\gamma = 0.5 * \frac{\text{Energy}(t)}{\text{Spermatheca size}} \quad (4)$$

If a drone is selected to mate with the queen, his sperm is added to the queen's spermatheca. After the queen finishes her mating flight, she returns to the nest and starts breeding by selecting sperm from her spermatheca at random and a new brood is created by crossover using the drone's and queen's genotypes. Workers then act on the broods as a set of mutation functions; therefore, if the same sperm is used once more to generate a brood, the resultant brood will be different because of mutation. Afterwards, the queen is replaced with the fittest brood if the latter is fitter than the former. The remaining broods are then killed and a new mating flight starts (Abbass, 2001a).

The flowchart and pseudo-code of the MBO algorithm are presented in Figs. 1 and 2, respectively.

The mating behaviour of honey bees has been considered than any other aspect of honey bee behaviour in creating evolutionary algorithms. As mentioned before, Abbass (2001a) presented the first search algorithm inspired by the mating process in honey bees. He used a single-queen, single-worker algorithm and applied this to a specific kind of assignment problem. Abbass (2001b) then presented a variation of the MBO algorithm where the colony contains a single queen with multiple workers, and used 6 different heuristic functions (workers) to generate and improve the solutions.

Abbass (2001c) also considered the honey bee colony with more than 1 queen in addition to a group of workers. Based on the experimental results, he concluded that a single queen was deemed better with an average number of broods. He also showed that more queens are useful when the number of broods is too low. Finally, the cooperative behaviour between the different heuristics was more functional than a single heuristic in isolation.

Teo and Abbass (2001) presented a modification of the MBO algorithm which could be considered as an extension of Abbass (2001a) and Abbass (2001c). The purpose of this modification was to use a more conventional annealing approach during the trajectory acceptance decision to guide the search process towards a more optimal solution space. New drones are only accepted as a potential source of sperm for mating if they are a more optimal drone; i.e., if the drone's genotype is fitter than the queen's. Otherwise, if it was a drone that took the search to a less optimal solution space, then it is only accepted probabilistically subject to the new annealing function. Teo and Abbass used 5 different heuristics for improving broods by workers. The experimental studies were conducted in 3 ways:

- testing each of 5 different heuristics working alone without MBO;
- testing the performance of each heuristic with the original MBO (Abbass, 2001a) and with the modified MBO; and
- testing the proposed algorithm against the original MBO using the five different heuristics operating in combination as a group of heuristics.

The results indicated that the algorithm's performance in the second group of experiments depended on the type of

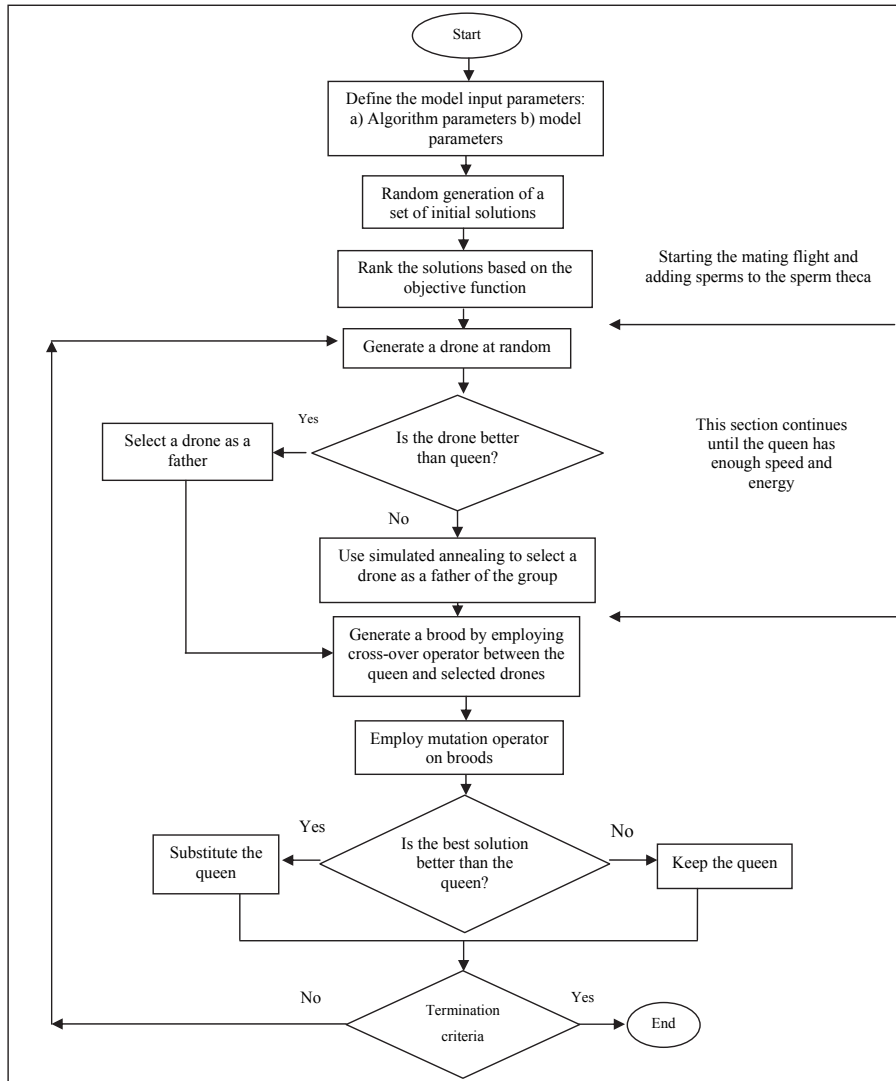


Figure 1
Computational flowchart and algorithm translation of natural processes

```

Start
Generate initial solutions at random
Select the best one as a queen
For a predefined number of mating flights
  Initialise energy and speed of the queen
  Generate a drone at random
  Compute energy reduction of the queen in each transition
  While the queen's energy > 0
    Evaluate the fitness of each drone
    If the drone is fitter than the queen and the queen's spermatheca are not full then
      add drone's sperm to queen's spermatheca
    Else queen probabilistically chooses drones
    Update energy of the queen
    Compute speed reduction of the queen in each transition
    Update speed of the queen
    With a probability of  $S(t)$  flip each gene in the drone's genotype
  End while
  For brood 1 = total number of broods
    Select sperm from the queen's spermatheca at random
    Generate a brood by crossing the queen's genome with the selected sperm
    Mutate the generated brood's genotype
  End for
  If the best brood is better than the queen then
    Replace the queen with the best brood
  Kill all broods
End for
End
  
```

Figure 2
Honey bee mating optimisation model's pseudo-code

heuristic function and the result of the modified algorithm is better than or similar to the original one. Finally, in the third group of experiments both annealing strategies were similarly efficient.

Teo and Abbass (2003) proposed another modification of MBO algorithm based on Teo and Abbass (2001), in which the previous drone is used as the basis for selecting the new drone. Moreover, from a biological point of view, the drone's creation is independent of the queen. Therefore, it is more natural to accept a sperm based on the drone's own fitness. Thus, in the modified algorithm the fitness of each drone is compared with the fitness of the previous one and the fitter drone is selected as a potential father of the group. Otherwise, the selection depends on the new simulated annealing function. In the new version of the simulated annealing function, the fitness of each drone is compared with the previous drone instead of the queen.

The performance of the modified algorithm was tested on 10 different assignment problems and compared with the performance of previous MBO algorithms. It was noticed that all heuristics failed to find even a single solution when working alone, whereas their performances were improved significantly when combined with the MBO. They found that the improved MBO algorithm outperformed the previous ones and was able to find solutions for problems where the previous versions had failed.

Chang (2006) presented the first demonstration of the capability of the MBO approach, from a theoretical perspective, for solving combinatorial optimisation problems and stochastic sequential decision making problems. The paper initially focused on the MBO algorithm for solving non-stochastic combinatorial optimisation problems and proved that the MBO has the ability to converge the global optimum value. The MBO was then used for solving stochastic dynamic programming (SDP) problems, and the algorithm was also proved converging to the optimal value. Chang points out that the MBO can be considered as a hybrid scheme of simulated annealing and genetic algorithms. From this perspective, simulated annealing corresponds to the queen's mating flight to obtain the potential drone's sperm in her spermatheca, and the genetic algorithm corresponds to the broods' generational improvement step, with some differences. The MBO algorithm has also been used

in different problems such as a clustering (Fathian et al., 2007) and scheduling (Koudil et al., 2007).

Haddad et al. (2006) developed an optimisation model based on the MBO algorithm proposed by Abbass (2001a, c) for finding an optimum operation policy for a single reservoir, and called it 'honey-bees mating optimisation' (HBMO). Later, Haddad et al. (2008) applied their HBMO model to extract the linear monthly reservoir operation rules for both irrigation and hydropower purposes. Sabbaghpour et al. (2012) used the honey bee mating optimisation algorithm for the calibration of the water distribution network of a town in the north of Iran, with promising results.

In this paper, we demonstrate the development and application of the MBO algorithm for long-term optimisation of multi-reservoir system operation. This is done in a comparative way to evaluate the status of MBO versus some other well-known meta-heuristic algorithms. A 3-reservoir system in Karkheh Basin, located in south-western Iran, was selected for this analysis, which consists of 47 years of monthly time steps (a total of 3 948 decision variables). It is worthwhile to mention that the long-term multi-reservoir operation problem of the Karkheh Basin system is a complex optimisation problem due to the large number of decision variables. In this case, the algorithm has to determine the optimum release from each reservoir and the optimum allocation to the agriculture regions, while at the same time preserving the minimum environmental in-stream flow after each diversion. Based on our experience in this system, most evolutionary algorithms face difficulty in finding even a near-optimum solution of the system and lack the ability to reach an acceptable solution in a reasonable amount of time. Therefore, this provides a good case study to verify the ability of any algorithm in solving large-scale problems.

CASE STUDY

In this study 3 reservoirs from Karkheh Basin were selected for evaluating the models' performance. These were: Sazban, Tangemashoore and Karkheh Reservoirs, located in Karkheh River basin in the south-west of Iran (Fig. 3).

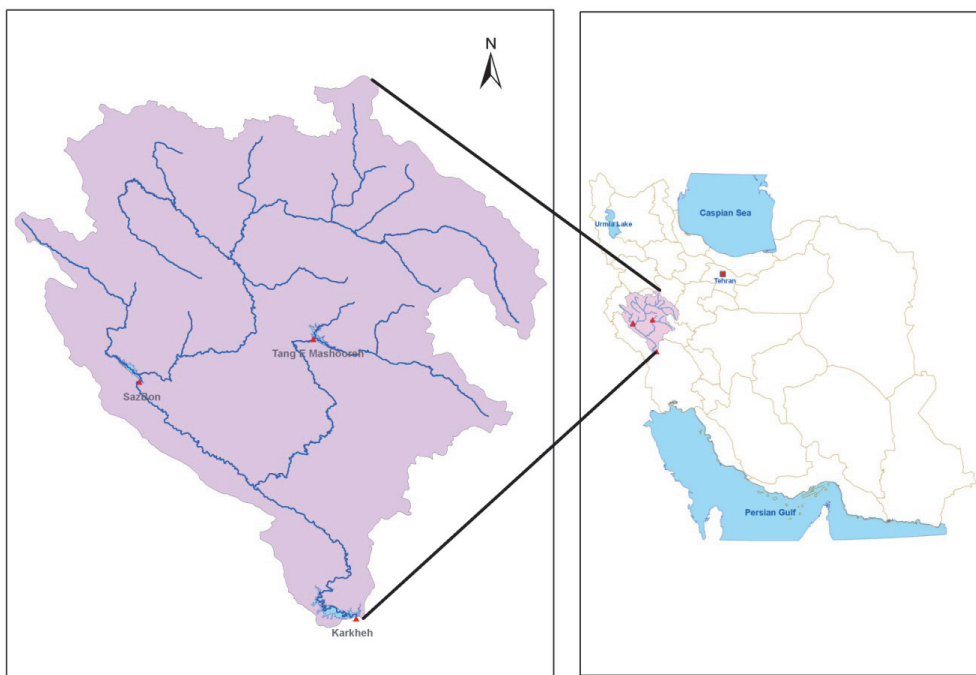


Figure 3
Location of Karkheh River basin system in Iran

Sazban Reservoir is located on Seimare River, an upstream main tributary of Karkheh River. Tangemashoore Reservoir is located within 90 km of the city of Khoramabad on the Kashkan River, another main tributary of Karkheh River. Karkheh Reservoir is located 20 km north-west of Andimeshk on the main Karkheh River in Khuzestan Province. The objective of Karkheh Reservoir is to control and regulate the flow of Karkheh River in order to provide irrigation water for downstream agricultural regions, generate energy from hydropower and control seasonal floods that may cause damage to downstream areas. At the same time, the reservoir must maintain a minimum in-stream flow in the Karkheh River for environmental purposes.

In this study, for simplicity, we have only considered water supply and environmental objectives. Therefore, in our models the reservoir system is operated for two main purposes: potable water supply for 4 agricultural regions and preservation of minimum environmental flows after each diversion. Since meeting the minimum flow is a preference in this system, we have used it as a strict constraint in our models. Nevertheless, it should be mentioned that there might be occasions when the system would not be able to fulfil this constraint due to extreme droughts. The rule is highly penalised when such a constraint violation occurs. Figure 4 illustrates the schematic of the Karkheh 3-reservoir system.

At first, we used a short-term 12-month period problem to see how the algorithms work in a simple multi-reservoir problem. Data for water year 1954–55 were used to represent an average year in the short-term problem. Reservoir storage at the end of each month and the allocations to each agricultural region are defined as the decision variables. Therefore, there are a total of 84 (12×7) decision variables in the short-term problem.

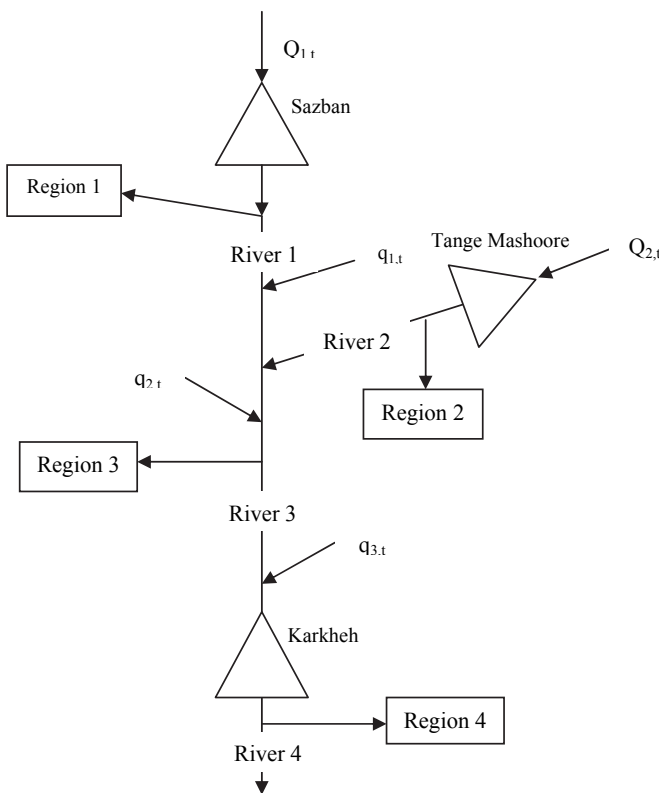


Figure 4
Schematic graph of Karkheh reservoir system

The objective function shown in Eq. (5) is defined as minimising the sum of squared deviation of releases from the target demands:

$$\text{Min } Z = \begin{cases} \sum_{t=1}^{12} \sum_{d=1}^4 (Rg_t^d - TD_t^d)^2 & \text{if } Rg_t^d < TD_t^d \\ 0 & \text{else} \end{cases} \quad (5)$$

Constraints are defined as follows:

$$S_{t+1}^{NR} = S_t^{NR} + Q_t^{NR} - (Rg_t^d + Rr_t^d) - E_t^{NR} \quad (6)$$

$$S_{T+1}^{NR} = S_1^{NR} \quad (7)$$

$$S_{\min}^{NR} \leq S_t^{NR} \leq S_{\max}^{NR} \quad (8)$$

$$Rr_t^d \geq Rr_{\min}^d \quad (9)$$

where:

S , Q , R and E are reservoir storage at the beginning of each time step, reservoir inflow, total reservoir release, and net reservoir evaporation loss, respectively, all in volumetric million-cubic-meter (mcm) units

NR refers to the reservoir number

Rg^d is the water allocation to Agricultural Region d

Rr^d is the remaining flow in the river after diversion for Region d

Rr^d indicates the streamflow releases by the upstream reservoirs to meet various demands at the downstream regions, including the minimum required flow in the river

Net evaporation from reservoirs is calculated by first estimating the mean water surface area in each period and then using the long-term monthly mean net evaporation rate to estimate the volume of net evaporative loss. Storage surface area in each reservoir is estimated through a simple linear regression as shown in Eq. (10).

$$A_t^{NR} = a * S_t^{NR} + b \quad (10)$$

where:

A_t^{NR} is reservoir area in km^2

S_t^{NR} is NR reservoir storage in mcm

The solution process in the MBO model starts with determining random decision variables in the feasible space. Then the outflow of each reservoir after agricultural diversion is calculated considering constraints 6 to 9. In this model, a penalty function is activated if the downstream demands are not satisfied. However, first an attempt is made to transmit infeasible variables into the feasible region by adjusting other variables in the system, before applying any penalty. If this process fails then a penalty is assigned to the objective function.

The feasibility of decision variables is related to the amount of the environmental flow after each diversion. In fact, variables which are less than the minimum environmental flow are considered as infeasible. Based on the mass balance equation, repairing the infeasible decision variables is carried out in three steps:

- decrease the reservoir storage at the end of each month to the minimum possible;
- increase the reservoir storage in each month to the maximum possible;
- decrease the allocation to the agricultural region to the minimum possible.

These three steps are consecutive and if the demand is met in one step, it is not necessary to run the next step. Also, if the

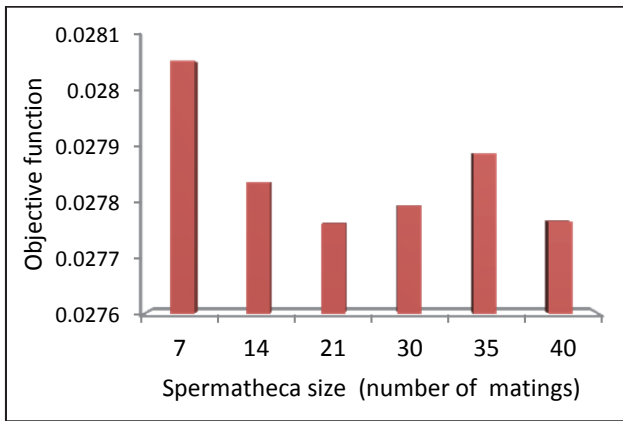


Figure 5
Objective function variations due to spermatheca size in short-term problem

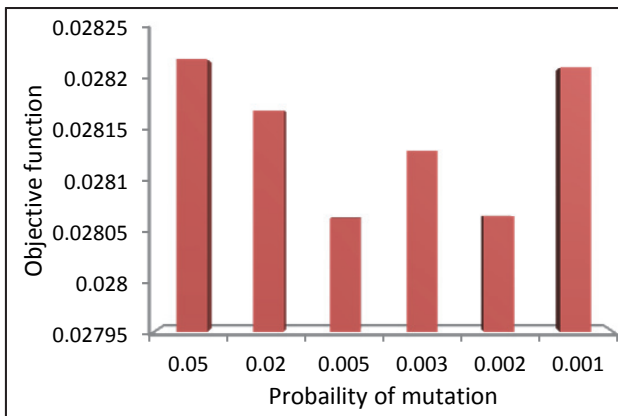


Figure 6
Objective function variations due to mutation rate in short-term problem

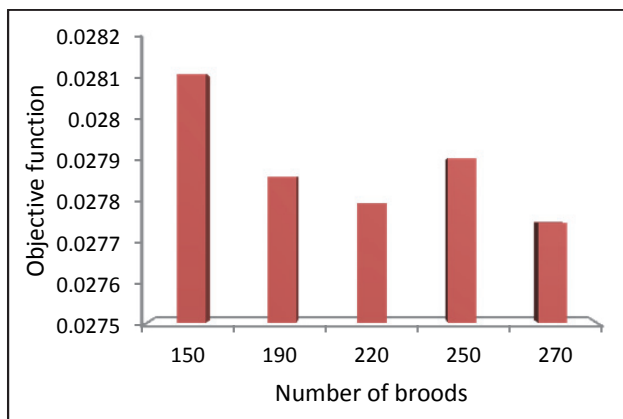


Figure 7
Objective function variations due to number of broods in short-term problem

Parameter	Value
Number of broods per flight	220
Spermatheca size (number of sperm)	21
Probability of mutation	0.005

environmental demand is not satisfied in any of these three steps, then the penalty function is activated in the objective function.

As mentioned earlier, the MBO algorithm has 3 user-defined parameters. These are: the number of broods born in each mating flight, the size of spermatheca (the maximum possible number of matings by the queen per mating flight), and the mutation rate. Sensitivity analysis has been carried out to determine the optimum value of these parameters and the results are presented in Figs 5 to 7 and Table 1.

As explained earlier, in the MBO algorithm the workers represent heuristic functions in which crossover functions are used to generate broods and mutation functions are used to improve their genotype. In the short-term model, the utilisation of simple crossover and mutation functions resulted in poor performance. But when these functions were randomly employed in the solution process the model performance was improved. Therefore, in this model the crossover operator is randomly selected from 1-point, 2-point, uniform and arithmetic crossovers, and the mutation operator is also randomly selected from uniform and boundary mutations.

The results of 10 independent runs and their comparison with the results from the other evolutionary algorithms are presented in Table 2.

Models based on ant colony optimisation for continuous domains (ACO_R), particle swarm optimisation (PSO), elitist-mutation particle swarm optimisation (EMPSO), and the genetic algorithm (GA) were used for comparison with the MBO algorithm (Dariane and Moradi, 2011). The characteristics of each of these methods are presented in Dariane and Moradi (2011) and are not included here.

Parameter	OF (mcm)				
	GA [†]	EMPSO [†]	PSO [†]	ACO _R [†]	MBO
Min.	27 620	28 745	30 331	28 705	27 663
Max.	27 820	31 097	35 384	29 592	27 863
SD	61	794	1 788	326	75
Average	27 720	29 514	33 281	29 144	27 745

[†]From Dariane and Moradi(2011)

As it can be seen from Table 2, the result of the MBO algorithm is better than those of ACO_R, PSO and EMPSO algorithms and it is almost equal to the result of genetic algorithm. The global optimum for this problem is 27 526 mcm, which is less than 0.8% different from the best result of the MBO algorithm. In the next step, we used the long-term problem of Karkheh system to evaluate the performance of MBO algorithm in a relatively large-scale problem. Long-term problem consists of 47 years of monthly periods and therefore considering the huge number of decision variables (i.e. 3 948 decision variables). Any algorithm applied to solve this problem would face a great challenge. Therefore, this provides a good medium to verify the ability of any algorithm in solving large-scale problems.

The objective function and all constraints are the same as those defined for the short-term problem. The only other alteration beside the extension of optimisation period is the inclusion of carry over year storage constraint in the long-term model. Sensitivity analysis similar to the short-term model is carried out to determine the optimum value of model parameters in the long term-models. The results are presented in Figs. 8 and 9 and Table 3.

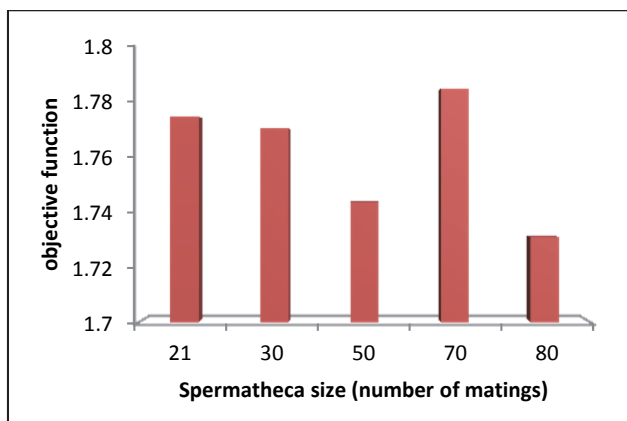


Figure 8
Objective function variations due to spermatheca size in long-term problem

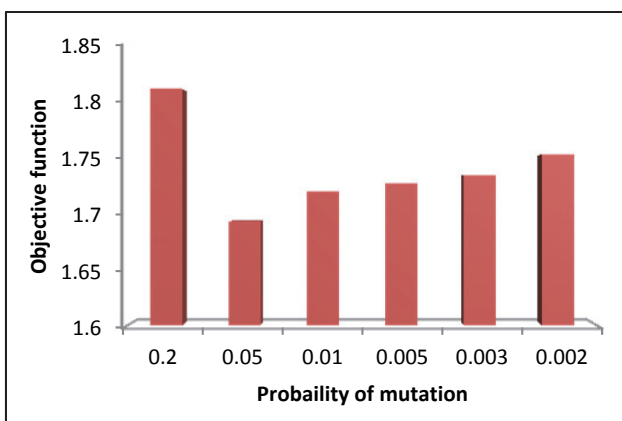


Figure 9
Objective function variations due to mutation rate in long-term problem

Parameter	Value
Number of broods per flight	60
Sperm theca size	50
Probability of mutation	0.05

In the long-term model, we first used the random selection approach for choosing crossover and mutation functions as in the short term model. But, because of the large number of decision variables, the solution was trapped in the infeasible region and the model was unable to find any feasible solution in the first 15 000 mating flights. Therefore, the extended intermediate crossover and BGA mutation methods, introduced by Muhlbein and Schlierkamp-Voosen (1993), are successfully applied in the long term problem. The results indicate that in the first 3 000 mating flights the solution is in the infeasible region and is gradually moving into the feasible region. The time interval of transmission to feasible region is different among runs.

For comparison purposes, the long term model is set to run for 2 000 s. The results of 10 independent runs are shown in Table 4.

The global optimum solution for this long-term problem is 0.525 units (to keep it simple each unit is set equal to 10^6 mcm). As it can be seen from Table 4, the MBO algorithm has reached

Parameter	OF (10^6 mcm)				
	GA [†]	EMPSO [†]	PSO [†]	ACO _R [†]	MBO
Min	4.95	3.1	5.4	4.7	1.67
Max	5.00	3.2	5.9	5.4	1.80
Stdev	0.02	0.04	0.22	0.29	0.03
Average	4.98	3.15	5.80	5.00	1.72

[†]From Dariane and Moradi (2011)

strictly better solution than any of other algorithms including GA. The average optimum objective function (OF) over 10 different runs for the MBO algorithm is equal to 1.72 units, and is about 1.2 units worse than the global optimum solution. The next best solution is achieved by the EMPSO algorithm and is equal to 3.15 units, which is nearly twice as far from the global optimum as the MBO. The GA has reached an average solution equal to 4.98 units over 10 runs and ACO_R and PSO also achieved to 5 and 5.8 units, respectively. It is clearly evident that there is a great gap between the solutions obtained by the MBO and other algorithms, which indicates a strict superiority of MBO over others. These results reveal the capacity of the MBO algorithm in conducting an extensive search in the entire search space of large-scale problems at the assumed time interval. However, since some algorithms behave differently in longer run time intervals, we decided to allow all the algorithms run for 4 h on the same machine. The results are presented in Table 5.

Parameter	Objective function (10^6 mcm)				
	GA [†]	EMPSO [†]	PSO [†]	ACO _R [†]	MBO
Min,	3.06	1.99	3.7	3.75	1.49
Max.	3.10	2.06	3.9	3.85	1.58
SD	0.02	0.03	0.08	0.04	0.04
Average	3.09	2.02	3.8	3.8	1.54

[†]From Dariane and Moradi (2011)

Again results shown in Table 5 reveal that the MBO algorithm continued to find better solutions as compared to those of other algorithms. In this time interval MBO reached a solution with objective function equal to 1.54 units, indicating a 10% improvement over the solution at the previous time step. Other methods also improved their results and achieved better solutions but were still far inferior when compared to the MBO. The performance of ACO_R, PSO, EMPSO and GA algorithms was improved by 24, 34, 36 and 38%, respectively. Though these algorithms displayed more progress toward the global solution in the longer time period, their final solutions were far inferior to that of the MBO. The EMPSO algorithm was still the second best performer with an average solution equal to 2.02. GA, ACO_R and PSO algorithms are ranked in order as the next best methods with the last two being very similar.

It can be concluded that the MBO algorithm has a high convergence rate at the beginning of the optimisation search process and performs better than any of the other algorithms tested here, in any run-time intervals. Although none of the algorithms was able to find the global optimum, MBO was able to get fairly good solutions for this multi-reservoir problem having a huge number of decision variables (i.e., 3 948). EMPSO also showed rapid convergence at the beginning of the search process and good

improvement with extension of the time interval. On the other hand, GA has a slow convergence at the beginning but proves to have very good improvement trend as run-time is extended.

Based on these observations, development of a hybrid algorithm consisting of MBO and any of the GA or EMP SO algorithms could be considered, in order to take advantage of their unique properties, such as rapid convergence in MBO and the improvement trend displayed by GA or EMP SO.

CONCLUSION

In this paper, the honey-bee mating optimisation algorithm is used to find the optimum short- and long-term operational path of Karkheh 3-reservoir system in south-western Iran. The short-term model has 84 decision variables and all algorithms reached a near-global optimum solution in a short amount of time due to the low number of decision variables. The MBO algorithm shows a markedly better performance than ACO_R, PSO and EMP SO, and a relatively similar performance to the GA.

The number of decision variables in the long-term model is 3 948. Because of this large number of variables, most of the known algorithms face a challenge in finding even a near-global optimum solution. Comparison between the results in different time intervals shows that the MBO algorithm has better functionality and much higher convergence in solving the long-term model than other evolutionary algorithms. MBO produced the best results in all time intervals and achieved results that were closer than any other algorithm to the global optimum solution. Test application of the algorithm revealed its capacity in solving large-scale water resource problems. Based on these observations, development of a hybrid algorithm consisting of MBO and any of the GA or EMP SO algorithms could be considered in future research to further aid in solving complex optimisation problems with a large number of decision variables.

REFERENCES

- ABBASS HA (2001a) A single queen single worker honey-bees approach to 3-SAT. *Proc. Genetic and Evolutionary Computation Conference (GECCO2001)*, 7–11 July 2001, San Francisco. 807–814.
- ABBASS HA (2001b) A pleometrosis MBO approach to satisfiability. *Proc. International Conference on Computational Intelligence for Modeling, Control and Automation, CIMCA2001*, 9–11 July 2001, Las Vegas, USA.
- ABBASS HA (2001c) Marriage in honey bees optimisation: A haplo-metrosis polygynous swarming approach. *Proc. Congress on Evolutionary Computation, CEC2001*, 27–30 May 2001, Seoul, Korea. 207–214.
- AZEEM MF and SAAD AM (2004) Modified queen bee evolution based genetic algorithm for tuning of scaling factors of fuzzy knowledge base controller. *Proc. India Annual Conference IEEE INDICON 2004*, 20–22 December, IIT Kharagpur, India. 299–303.
- BASTURK B and KARABOGA D (2006) An artificial bee colony (ABC) algorithm for numeric function optimisation. *Proc. IEEE, Swarm Intelligence Symposium*, 12–14 May 2006, Indianapolis, Indiana, USA.
- BONABEAU E, DORIGO M and THERAULAZ G (1999) *Swarm Intelligence: From Natural to Artificial Intelligence*. Oxford University Press, New York, USA.
- CHANG HS (2006) Converging marriage in honey-bees optimisation and application to stochastic dynamic programming. *J. Global Optim.* 35 423–441.
- DARIANE AB and MORADI AM (2010) Application of ant colony based algorithm to multi reservoir water resource problems. *J. Water Wastewater* 76 (4) 81–91 (in Farsi).
- DIOWLD K, BEEKMAN M and MIDDENDORF M (2010) Bee nest site selection as an optimisation process. *Proc. Alife XII Conference*, 19–23 August 2010, Odense, Denmark. 626–633.
- FATHIAN M, AMIRI B and MAROOSI A (2007) Application of honey-bee mating optimisation algorithm on clustering. *J. Appl. Math. Comput.* 190 (2) 1502–1513.
- HADDAD OB, AFSHAR A and MARIANO MA (2008) Design-operation of multi-hydropower reservoirs: HBMO approach. *J. Water Resour. Manage.* 22 (12) 1709–1722. DOI: 10.1007/s11269-008-9249-5.
- HADDAD OB, AFSHAR A and MARIANO MA (2006) Honey-bees mating optimisation (HBMO) algorithm: A new heuristic approach for water resources optimisation. *J. Water Resour. Manage.* 20 (5) 661–680.
- KARABOGA D (2005) An idea based on honey bee swarm for numerical optimisation. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey.
- KOUDIL M, BENATCHBA K, TARABET A and SAHRAOUI EB (2007) Using artificial bees to solve partitioning and scheduling problems in codesign. *J. Appl. Math. Comput.* 186 (2) 1710–1722.
- LEMMENS NPPM (2006) To bee or not to bee: A comparative study in swarm intelligence. Masters thesis, Maastricht University, Maastricht ICT Competence Centre, Institute for Knowledge and Agent Technology, Maastricht, The Netherlands.
- LUCIC P and TEODOROVIC D (2001) Bee system: Modeling Combinatorial optimisation transportation engineering problems by swarm intelligence. Preprints of the *TRISTAN IV Triennial Symposium on Transportation Analysis*, 13–19 June 2001, Sao Miguel, Azores Islands. 441–445.
- MORADI AM (2009) Optimisation of multi-reservoir systems using meta-heuristic algorithms. M.Sc. thesis, K.N. Toosi University of Technology (in Farsi).
- MÜHLENBEIN H and SCHLIERKAMP-VOOSEN D (1993) Predictive models for the breeder genetic algorithm: I. Continuous parameter optimisation. *Evol. Comput.* 1 (1) 25–49.
- PHAM DT, KOG E, GHANBARZADEH A, OTRI S, RAHIM S and ZAIDI M (2006) The bees algorithm: A novel tool for complex optimisation problems. *Proc. 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006)*, 3–14 July 2006. Elsevier, Oxford. 454–459.
- RINDERER TE and COLLINS AM (1986) Behavioral genetics. In: Rinderer TE (ed.) *Bee Genetics and Breeding*. Academic Press, Inc., Orlando. 155–176.
- SABBAGHPOUR S, NAGHASHZADEGAN M, JAVAHERDEH K and HADDAD O (2012) HBMO algorithm for calibrating water distribution network of Langarud city. *Water Sci. Technol. J.* 65 (2) 1564–1569.
- SATO T and HAGIWARA M (1997) Bee system: Finding solution by a concentrated search. *Proc. IEEE International Conference on Systems, Man and Cybernetics*, 12–15 October 1997, Hyatto Rlandoo, Rlndof, Lordia, USA. 3954–3959.
- SCHMICKL T, THENIUS R and CRAILSHEIM K (2005) Simulating swarm intelligence in honey bees: Foraging in differently fluctuating environments. *Proc. 2005 Conference on Genetic and Evolutionary Computation*, 25–29 June 2005, Washington DC. 273–274.
- SEELEY TD and BUHRMAN SC (1999) Group decision making in swarms of honey bees. *J. Behav. Ecol. Sociobiol.* 45 19–31.
- TEO J and ABBASS HA (2001) An annealing approach to the mating-flight trajectories in the marriage in honey bees optimisation algorithm. Technical Report CS04/01. School of Computer Science, University of New South Wales at Australian Defence Force Academy (ADFA).
- TEO J and ABBASS HA (2003) A true annealing approach to the marriage in honey-bees optimisation algorithm. *Int. J. Comput. Intell. Appl.* 3 (2) 199–208.
- YANG XS (2005) Engineering optimisations via nature-inspired virtual bee algorithms. In: Yang JM and Alvarez JR (eds.) *IWINAC 2005, Lecture Notes in Computer Science (LNCS) 3562*, Springer-Verlag, Berlin. 317–323.
- YONEZAWA Y and KIKUCHI T (1996) Ecological algorithm for optimal ordering used by collective honey bee behavior. *Proc. 7th International Symposium on Micro Machine and Human Science*, 2–4 October 1996, Nagoya. 249–256.