BSA: A Hybrid Bees' Simulated Annealing Algorithm To Solve Optimization & NP-Complete Problems

Dr. Ahmed Tariq Sadiq *& Amaal Ghazi Hamad

Received on: 4/8/2008
Accepted on: 4/6/2009

Abstract

Swarm-based algorithms such as Bees Algorithm BA have proven to be very powerful computational techniques due to their search capabilities. Other methods which are useful in diverse application areas are simulated annealing, evolution strategies etc. The searching ability of these algorithms can be improved by properly blending their characteristic features. In this paper an attempt is made to intermix the search properties of BA and SA, in order to develop a hybrid algorithm which is equally applicable and has a better searching ability and power to reach a near optimal solution. This leads to the development of a fast method to solve complicated types of optimization and NP-complete problems.

خوارزمية النحل ذات التلدين المقلد الهجينة لحل الامثلية ومشاكل من نوع NP-Complete

الخلاصة

الخوارزميات المعتمدة على الحشد مثل خوارزميات النحل تم اعتمادها كتقنيات أحتسابية ذات قوة جيدة في مهمات البحث. ومن الطرق الاخرى المفيدة تطبيقات التنوع هي التلدين المقلد واستراتيجيات التطور. قابلية البحث لهذه الخوارزميات ممكن تطويرها من خلال مزج صفاتها الجيدة فيما بينها. في هذا البحث سنقدم خوارزمية مهجنة لها صفات خوارزمتي النحل والتلدين المقلد، ولكي نطور الخوارزمية المهجنة يجب ان تضمن لنا قابلية البحث الافضل والوصول الى أفضل حل أو أقرب اليه. هذا بدوره سيقود الى تطوير طرق سريعة لحل الانواع المعقدة من مشاكل الامثلية المعقدة ومن نوع NP-Complete.

1. Introduction

Many complex multi-variable optimization problems cannot be solved exactly within polynomially bounded computation times. This generates much interest in search algorithms that find near-optimal solutions in reasonable running times. The Bees Algorithm is an optimization algorithm that mimics the food foraging behaviour of swarms of honey bees. The algorithm has been Successfully applied to different problems including the training of neural networks for pattern recognition [15], the formation of homogeneous data clusters [16] and the generation of multiple feasible solutions to a preliminary design problem [17].

Simulated Annealing (SA), is a randomized optimization technique which simulates the metallurgical cooling process. This technique avoids

*Computer Sciences Department, University of Technology /Baghdad

https://doi.org/10.30684/etj.28.2.5
2412-0758/University of Technology-Iraq, Baghdad, Iraq

the problem of getting stuck in a local optimum and leads to the global optimum solution.

Section 2 reviews related work in the area of intelligent optimization. Section 3 describes the foraging behaviour of natural bees and the core ideas of the Bees Algorithm. Section 4 presents basics of SA. In Section 5 the proposed hybrid algorithm is detailed. In Section 6 two test problems are showed . Section 7 details the results of applying both the original and the modified BA. Section 8 concludes the paper.

2.IntelligentSwarm-based Optimization

Swarm-basedoptimization algorithms (SOAs) mimic nature's methods to drive a search towards the optimal solution. The difference between SOAs and direct search algorithms is that SOAs use a population of solutions for every iteration instead of a single solution. The outcome of each iteration is also a population of solutions. SOAs include the Ant Colony Optimization (ACO) algorithm [6], the Genetic Algorithm (GA) [7] and the Particle Swarm Optimization (PSO) algorithm [8]. Common to all population-based search methods is a strategy that generates variations of the solution being sought. A very successful non-greedy population-based algorithm is the ACO algorithm which emulates the behaviour of real ants. Ants are capable of finding the shortest path from the food source to their nest using a chemical substance called pheromone to guide their search. The pheromone is deposited on the ground as the ants move and the probability that a passing stray ant will follow this trail depends on the quantity of pheromone laid. ACO was first used for functional optimization by Bilchev [9] and further attempts were reported in [1, 9, and 10].

The Genetic Algorithm is based on natural selection and genetic recombination. The algorithm works by choosing solutions from the current population and then applying genetic operators – such as mutation and crossover – to create a new population. The algorithm efficiently exploits historical information to speculate on new search areas with improved performance [7]. When applied to optimization problems, the GA has the advantage of performing global search. Particle Swarm Optimization (PSO) is an optimization procedure based on the social behavior of groups of organizations, for example the flocking of birds or the schooling of fish [8]. Individual solutions in a population are viewed as "particles" that evolve or change their positions with time. Each particle modifies its position in search space according to its own experience and also that of a neighboring particle by remembering the best position visited by itself and its neighbors, thus combining local and global search methods [1,8]. There are other SOAs with names suggestive of possibly bee-inspired operations [11, 12, 13, and 14].

3. The Bees Algorithm

3.1. Bees in nature

A colony of honey bees can extend itself over long distances to exploit a large number of food sources [11,12]. A colony prospers by deploying its foragers to good fields[1,13,14].

The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees move randomly from one patch to another[1,12].

When those scout bees found a patch which is rated above a certain quality threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the "waggle dance" [11].

This mysterious dance is essential for colony communication, and contains three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating (or fitness) [1,11,14]. This information helps the colony to send its bees to flower patches precisely, without using guides or maps[14]. After waggle dancing on the dance floor, the dancer (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. More follower bees are sent to more promising patches. This allows the colony to gather food quickly and efficiently [1, 14].

3.2. Bees Algorithm [1]

As mentioned, the Bees Algorithm is an optimization algorithm inspired by the natural foraging behaviour of honey bees to find the optimal solution [8]. Figure 1 shows the pseudo code for the algorithm in its simplest form. The algorithm requires a number of parameters to be set, namely: number of scout bees (*n*), number of sites selected out of n visited sites (*m*), number of best sites out of m selected sites (*e*), number of bees recruited for best e sites (*nep*), number of bees recruited for the other (*m-e*) selected sites (*nsp*), initial size of patches (*ngh*) which includes site and its neighborhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of the sites visited by the scout bees are evaluated in step 2.

In step 4, bees that have the highest fitnesses are chosen as "selected bees" and sites visited by them are chosen for neighborhood search. Then, in steps 5 and 6, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitnesses associated with the sites they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm.

However, in step 6, for each patch only the bee with the highest fitness will be selected to form the next bee population. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population–representatives from each selected patch and other scout bees assigned to conduct random searches.

4. Concept of Simulated Annealing

SA, is an optimization procedure that performs randomized search in large, complex and multimodal search space for providing a near optimal solution. Annealing is a metallurgical process where the ground state behavior of a metal is studied by gradually changing the substance from a molten state to the thermo dynamical lowest energy state. In simulated annealing a problem state is defined by the values of a number of parameters. The state transition is done by changing the values of the parameters using the Boltzmann distribution function in thermodynamics. The objective is to minimize the value of some objective function. At each state transition the temperature of the system is reduced by a small amount.

The temperature schedule is designed so that the state of the system freezes after hundreds of transitions. A logarithmically decreasing temperature is found useful for convergence without getting stuck to a local minimum state. But to cool down the system to the equilibrium state it takes time. In particular, simulated annealing knows little about whether a region of the search space has been explored or whether a region is better for searching by the use of statistical distribution function [3,18].

In simulated annealing any optimization (minimization is considered here) problem can be treated as a pair (*S*, *f*) where *S* R^n and *f* :S→R is an *n*-dimensional real valued bounded function. Objective is to find x_{min} . S such that f ($x_{min)}$ is global minimum value of the bounded function f.

In other words, it is required to find out x_{min} . S such that for all x . S: *f* (x_{min}) ≤ *f*(x) . The initial configuration x_0 is generated randomly or from some domain specific information. At each step an alternative configuration y is selected from a set of alternatives. The configuration y is accepted if *f* (y) ≤ *f*(x) otherwise it is be accepted with probability exp [-(*f* (y)-*f*(x))/T] .

At each iteration of the process, the temperature T, is reduced by a small amount. The whole process is repeated until a near optimal solution is found. A set of control parameters govern the convergence of the algorithm. These parameters are initial temperature T_0, a decrement function for decreasing the value of T at every iteration, a lower limit for T or the maximum number of iterations needed for convergence.

5. The Proposed Hybrid Bees' Simulated Annealing Algorithm

The new variant of BA called Bees' Simulated Annealing Algorithm BSA requires the same parameters of the original BA ,and the parameters of the SA.

The algorithm starts with *n* scout bees randomly distributed in the search space. The fitness of the sites (i.e. the performance of the candidate solutions) visited by the scout bees are evaluated in step 2. In step 4, the m sites are designated as "selected sites" and chosen for neighborhood search. In Step 5, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitnesses associated with the sites they are visiting.

In Basic Bees version, in Step 6 for each patch only the bee with the highest fitness will be selected to form the next bee population.

However, in Bees' Simulated Annealing version , representative bee for each patch is chosen based on Boltzmann probability distribution function. Let it be assumed that f_{min} is the fitness of the currently available representative bee . If the next has fitness f (yi) such that f (yi) < f_{min}, then the new string is selected as the representative bee otherwise it is selected with Boltzmann probability P
P=exp[-(fitness(yj)-fitness(xj))/T]...(1)
where T = T_0 $(1-\alpha)^k$,k is the iteration number , α is the decrement value taken from the range [0,1] and T_0 from the range [5,100] . Equation (1) shows that the value of T decreases exponentially or at logarithmic rate with increase in the value of k and hence the value of the probability P. In Step 7 , the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions.

At the end of each iteration, the colony has two parts to its new population: representatives from the selected patches, and scout bees assigned to conduct random searches. These steps are repeated until a stopping criterion is met.

6. Experimental problems

Two standard optimization problems were used to test the Bees' Simulated Annealing Algorithm and to compare it's performance with the original algorithm.

6.1: 4 Color Map Problem [4]

The celebrated 4 Color Map Theorem_states that any map in the plane or on the sphere can be colored with only four colors such that no two neighboring countries are of the same color. The problem has a long history and inspired many people (including many non-mathematicians and in particular countless high school students) to attempt a solution.

The proof of the four color theorem by Haken and Appel [19] was so involved it required computational support to complete. It is well known that determining if a graph can be colored by a certain number of colors is NP-complete, but it is also known that even approximating the chromatic number of a graph is NP-hard [20].

Many algorithms have been applied to this problem, such as simulated annealing, Tabu search [22], genetic algorithms [21], neural networks [23], and hybrid algorithms such as Paquete's iterated and Tabu local search algorithm [24]. Although some swarm based algorithms have been applied to graph coloring such as Comellas' [25] and Costa's [26], algorithms of this kind have not been as heavily studied in the literature. There exist two main categories of algorithms: *successive augmentation algorithms* [27], which color a graph one vertex at a time, disallowing vertices from being re-colored and *iterative improvement algorithms*, which allow backtracking and re-coloring [29].

Each bee is a vector (N) ,where N is the number of cities in the map. An adjacency array of dimension NxN is used to identify the neighborhood of adjacent cities. The neighborhood search operator used is simply swapping two randomly chosen points.

6.2: ClassicalTransportation roblem

The classical transportation model

seeks the determination of a transportation plan of a *single* commodity from a number of sources to a number of destinations. The data of the model include[5]:

1. Level of *supply* at each source and amount of demand at each destination.
2. The *unit* transportation cost of the commodity from each source to each destination.

Since there is only one commodity, a destination can receive its demand from one or more sources. The objective *of* the model is to determine the amount to be shipped from each source to each destination such that the total transportation cost is minimized [5].

Let xij represent the amount transported from source *i* to destination *j*, then the model representing the transportation problem is given generally as[5]:

$$\text{minimize } z = \sum_{i=1}^{m}\sum_{j=1}^{n} cij\ xij$$

subject to

$$\sum_{j=1}^{n} xij \le ai \qquad i = 1,2,\ldots.,m$$

$$\sum_{i=1}^{m} xij \ge bi \qquad i = 1,2,\ldots..,n$$

$$xij \ge 0 \qquad \text{for all i and j}$$

Where ai is the amount of supply at source i, bj is the demand at destination j and cij is the unit transportation cost between source i and destination j[5] .

The first set of constraints stipulates that the sum of the shipments from a source cannot exceed its supply, similarly, the second set requires that the sum of the shipments to a destination must satisfy its demand [5].

A more compact method for representing the transportation model is to use what we call the **transportation tableau** (figure 4)**.** It is a matrix form with its rows representing the sources and its columns the destination. The cost elements c_{ij} are summarized in the northeast corner of the matrix cell *(i, j)* [5].

The neighborhood search for classical transportation problem is done by using a modification to the stepping stone method. Figure 5 illustrates the new procedure.

7. Results

For Bees and Bees' Simulated Annealing algorithms the same empirically values for the control parameters have been used. Those features with the additional BSA parameters are shown in table 1. Results of average 10 independent runs for both BA and BSA have proved that both algorithms are powerful technique capable of finding solutions close to the optimum. Results indicates that BSA has a faster convergence than the original BA.

Experiments shows that BSA has better performance for the four color map problem with $T_0 = 5$ and $\Delta t = 0.9$, while for the classical transportation problem $T_0 = 5$ and $\Delta t = 0.7$.Those results are shown in figures 6,7 respectively.

8. Conclusion

This paper proposes a hybrid Bees algorithm framework. This new approach is highly general and can be modified to suit any application area. Results of experiments with the selected problems show the applicability of the proposed method. The new hybrid algorithm can be consider as computationally fast optimizer tool without any special domain information. The new algorithm is found to be very beneficial in solving NP-complete problems. One of the drawbacks of the algorithm is the number of tunable parameters used. However, it is possible to set the parameter's values by conducting a small number of trials.

References

[1] Pham D.T., Ghanbarzadeh A., Koç E., Otri S., Rahim S., and Zaidi M. *The Bees Algorithm, A Novel Tool for Complex Optimisation Problems*. *Proc 2nd Virtual International Conference on Intelligent Production Machines and Systems*. 2006, Elsevier (Oxford), pp. 454-459.

[2] Pham D.T., Otri S., Haj Darwish A.. *Application of the Bees Algorithm to PCB assembly optimisation.*

[3] Pakhira M. K..*A Hybrid Genetic Algorithm using Probabilistic Selection.* Kalyani Government Engineering College. IE (I) Journal.CP ,Vol 84, May 2003.

[4] Peter Alfeld. *Bivariate Splines and the Four Color Map Problem*. College of Science ,University of Utah . 1997.

[5] Handy A. Taha. *Operations Research: An Introduction* .3rd edition. Collier Macmillan International Editions.

[6] Dorigo M and Stützle T. *Ant Colony Optimization*. MIT Press, Cambridge, 2004.

[7] Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading:Addison-Wesley Longman, 1989.

[8] Eberhart, R., Y. Shi, and J. Kennedy, *Swarm Intelligence*. Morgan Kaufmann, San Francisco, 2001.

[9] Bilchev G and Parmee IC. *The Ant Colony Metaphor for Searching Continuous Design Spaces.* in Selected Papers from AISB Workshop on Evolutionary Computing. (1995) 25 -39.

[10] Mathur M, Karale SB, Priye S, Jayaraman VK and Kulkarni BD. *Ant Colony Approach to Continuous Function Optimization.* Ind. Eng. Chem. Res.39(10) (2000) 3814-3822.

[11] Von Frisch K. *Bees: Their Vision, Chemical Senses and Language.* (Revisededn) Cornell University Press, N.Y., Ithaca, 1976.

[12] Seeley TD. *The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies.* Massachusetts: Harvard University Press, Cambridge, 1996.

[13] Bonabeau E, Dorigo M, and Theraulaz G. *Swarm Intelligence: from Natural to Artificial Systems.* Oxford University Press, New York, 1999.

[14] Camazine S, Deneubourg J, Franks NR, Sneyd J, Theraula G and Bonabeau E. *Self-Organization in Biological Systems.* Princeton: Princeton University Press, 2003.

[15] Pham D.T., Otri S., Ghanbarzadeh A., and Koc E. *Application of the*

Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition. Proc 2nd IEEE Information and Communication Technologies. 2006. Damascus, Syria, pp. 1624 - 1629.

[16] Pham D.T., Otri S., Afify A., Mahmuddin M., and Al-Jabbouli H. *Data Clustering Using the Bees Algorithm.* Proc 40th CIRP International Manufacturing Systems Seminar. 2007, Liverpool, UK

[17] Pham D.T., Castellani M., and Ghanbarzadeh A. *Preliminary Design Using the Bees Algorithm.* Proc 8th International Conference on Laser Metrology, CMM and Machine Tool Performance, LAMDAMAP. 2007. Cardiff, UK: Euspen, pp. 420-429.

[18] S Kirkpatrik, C Gellat and M P Vecchi. *.Optimization by Simulated Annealing.* Science, vol 220, 1983, pp 671-680.

[19] Wilson, Robin. *Four Colors Suffice,* (Princeton: Princeton University Press, 2000).

[20] Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1977.

[21] Filho, Geraldo and Luiz Antonio Nogueira Lorena. *Improvements on Constructive Genetic Approaches To Graph Coloring.* http://www.lac.inpe.br/~lorena/ifors99/agc-col-clus.pdf .2005.

[22] Fotakis, Dimitris and Spiridon D. Likothanassis, *An Evolutionary Annealing Approach to Graph Coloring,*http://www.tik.ee.ethz.ch/~stefanak/pubs/evocop2001.pdf .2005.

[23] Harmanani, Haidar, Jean Hannouche, and Nancy Khoury, *A Neural Networks Algorithm for the Minimum Coloring Problem Using FPGAs*. IASTED Internation Conference Applied Simulation and Modelling 12: 152-156.

[24] Luis Paquete and Thomas Stuzle, *an Experimental Investigation of Iterated Local Search for Coloring Graphs*. http://iridia.ulb.ac.be/~meta/newsite/downloads/EvoCOP02.LT.pdf

[25] Comellas, Francesc and Javier Ozon, *An Ant Algorithm for the Graph Colouring Problem*. "From Ant Colonies to Artificial Ants: 1st International Workshop on ACO", Belgium.

[26] D. Costa, and A. Hertz. *Ants Can Colour Graphs*. Journal of the Operational Research Society: 295-305.

[27] Gary Lewandowski and Anne Condon. *Experiments with Parallel Graph Coloring Heuristics and Applications of Graph Coloring.* DIMACS Series in Discrete Mathematics (1994).

[28] F. T. Leighton. *A Graph Coloring Algorithm for Large Scheduling Problems*. Journal of Research of the National Bureau of Standards: 489.

[29] Daniel Palmer, Marc Kirschenbaum, Jason Shifflet, Linda Seiter. *Swarm Reasoning.* Mathematics and Computer Science Department. John Carroll University

Table (1) Parameters of the BA and BSA

Parameters	Symbol	Value
Population size	N	60
Number of selected sites	M	30
Number of elite sites	E	15
Number of recruited bees for best m sites	nsp	3
Number of recruited bees for best e sites	N ep	5
Initial temperature	t_0	10,5
Decrement rule value	Δt	0.95-0.7
Number of iterations	itr	100

```
1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met)
//Forming new population.
4. Select sites for neighborhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While.
```

Figure (1) Pseudo code of the basic bees algorithm

```
Begin
Initialize (T, x)
Termination condition = false
While (NOT termination condition) do
    Begin
     For i = 1 to L do
        Begin
          Generate y from x
          if $f(y)-f(x) \le 0$
            Then x = y
                    Else if $\exp[-(f(y)-f(x))/T] >$ random [0, 1]
                                  Then x = y
        End
        Lower T
    End
End
```

Figure (2) Simulated Annealing Algorithm

```
1. Initialize population n with random solutions, .Initialize the initial temperature T0
2. Evaluate fitness of the population n.
3. While (stopping criterion not met)
//Forming new population.
4. Select sites m for neighborhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. for each patch j  compare best recruited bee yj with the bee recruited it xj
            if fitness(yj) - fitness (xj)  < 0 then
              xj :=yj
            else
             if  exp[-( fitness (yj)-fitness (xj))/T]  >random [0,1]
               then   xj: =yi
7.Assign remaining bees to search randomly and evaluate their fitnesses.
8. Decrement T;
9. End While.
```

Figure (3) Pseudo code of Bees' Simulated Annealing algorithm

11	10	0	20	15
20	12	7	9	25
18	0	14	16	5
5	15	15	10	

Figure (4) Transportation Tableau

1.randomly determine an entering variable from among the nonbasic variables.
2.randomly determine a leaving variable from among the basic variables.
3.determine a closed path starting and ending at the entering variable (the closed path contains only nonbasic variables and has basic variables corners).
4.begin from the entering variable assign '+' and alternate '-' at the corner squares.
5.detrmine the closed path basic '-' signed variable with the least assignment value v.
6.transfer allocation along the closed path by summing v to each '+' signed variable and subtracting v from each '-' signed variable.

Figure (5) Modified Stepping Stone

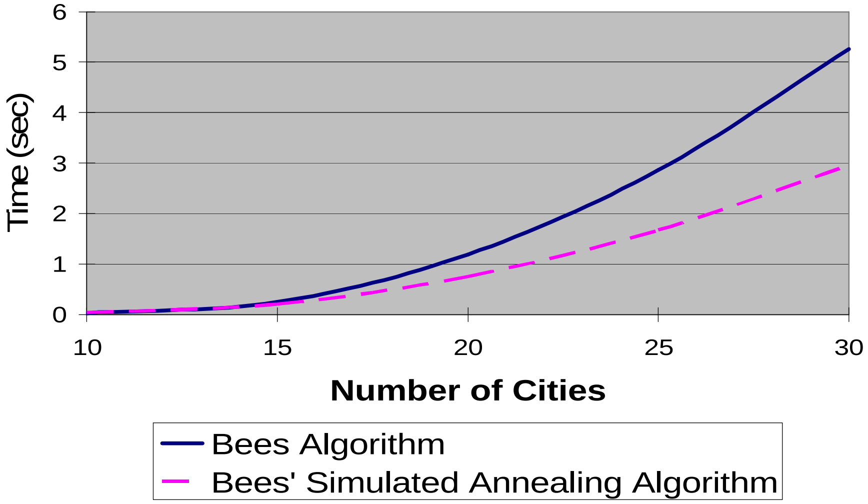

Figure (6) Performance of BA & BSA in 4_color map problem at T_0=5 & Δt = 0.9

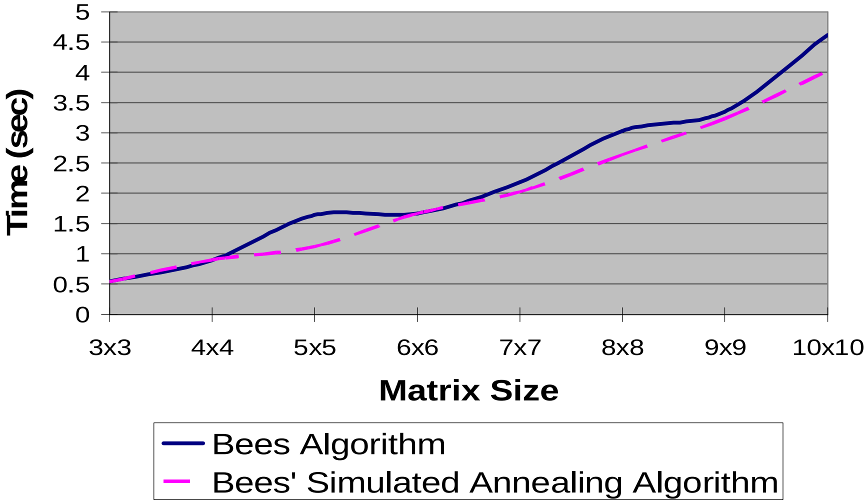

Figure (7) Performance of BA & BSA in classical transportation problem at T_0=5 & Δt=0.7