

Alternative Computing Summative Assignment

Dr Eleni Akrida and Dr Barnaby Martin

2020

Quantum Computing Assignment: Dr Eleni Akrida

Part A:

1. Which of the following pairs of expressions represent the same quantum state?
Justify your answers.

[25 marks]

- (a) $\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$ and $|0\rangle$.

$$\begin{aligned}\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) &= \frac{1}{\sqrt{2}} \cdot \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle + \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \right) \\ &= \frac{1}{\sqrt{2}} \cdot \left(\frac{2}{\sqrt{2}}|0\rangle \right) \\ &= \frac{2}{2} \cdot |0\rangle \\ &= |0\rangle\end{aligned}$$

These expressions therefore represent the same quantum state.

- (b) $i|1\rangle$ and $\frac{1+i}{\sqrt{2}}|1\rangle$.

As quantum states are directions, the state represented by $|1\rangle$ does not change when multiplied by (complex) constant. These expressions therefore represent the same quantum state.

- (c) $\frac{1}{\sqrt{2}}(|-\rangle + |+\rangle)$ and $\frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle)$.

$$\begin{aligned}\frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle) &= \frac{1}{\sqrt{2}} \cdot \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle + \frac{1}{\sqrt{2}}|0\rangle - \frac{-i}{\sqrt{2}}|1\rangle \right) \\ &= \frac{1}{\sqrt{2}} \cdot \left(\frac{2}{\sqrt{2}}|0\rangle \right) \\ &= |0\rangle \\ &= \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \quad \text{(as proven in part a))}\end{aligned}$$

These expressions therefore represent the same quantum state.

- (d) $\left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle\right)$ and $\left(\frac{\sqrt{6}+\sqrt{2}}{4}|+\rangle + \frac{\sqrt{6}-\sqrt{2}}{4}|-\rangle\right)$.

$$\begin{aligned}\frac{\sqrt{6}+\sqrt{2}}{4}|+\rangle + \frac{\sqrt{6}-\sqrt{2}}{4}|-\rangle &= \frac{\sqrt{6}+\sqrt{2}}{4\sqrt{2}}(|0\rangle + |1\rangle) + \frac{\sqrt{6}-\sqrt{2}}{4\sqrt{2}}(|0\rangle - |1\rangle) \\ &= \frac{(\sqrt{3}+1)|0\rangle + (\sqrt{3}+1)|1\rangle + (\sqrt{3}-1)|0\rangle - (\sqrt{3}-1)|1\rangle}{4} \\ &= \frac{2\sqrt{3}|0\rangle + 2|1\rangle}{4} \\ &= \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle\end{aligned}$$

These expressions therefore represent the same quantum state.

(e) $\frac{1}{\sqrt{2}}(|i\rangle - |-i\rangle)$ and $|0\rangle$.

$$\begin{aligned}\frac{1}{\sqrt{2}}(|i\rangle - |-i\rangle) &= \frac{1}{2}(|0\rangle + i|1\rangle - |0\rangle - i|1\rangle) \\ &= i|1\rangle \\ &\neq |0\rangle\end{aligned}$$

These expressions therefore do not represent the same quantum state.

2. For each state and measurement basis, describe the possible outcomes of a measurement of that state with respect to that basis and give the probability of each outcome. **[25 marks]**

(a) $(\frac{3i}{4}|+\rangle - \frac{\sqrt{7}}{4}|-\rangle)$ and $\{|0\rangle, |1\rangle\}$.

$$\begin{aligned}\frac{3i}{4}|+\rangle - \frac{\sqrt{7}}{4}|-\rangle &= \frac{3i}{4\sqrt{2}}(|0\rangle + |1\rangle) - \frac{\sqrt{7}}{4\sqrt{2}}(|0\rangle - |1\rangle) \\ &= \frac{-\sqrt{7} + 3i}{4\sqrt{2}}|0\rangle + \frac{\sqrt{7} + 3i}{4\sqrt{2}}|1\rangle\end{aligned}$$

The probability of this outcome for $|0\rangle$:

$$|\frac{-\sqrt{7}}{4\sqrt{2}} + \frac{3}{4\sqrt{2}}i|^2 = \frac{1}{2}$$

The probability of this outcome for $|1\rangle$:

$$|\frac{\sqrt{7}}{4\sqrt{2}} + \frac{3}{4\sqrt{2}}i|^2 = \frac{1}{2}$$

(b) $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and $\{|i\rangle, |-i\rangle\}$.

$$\begin{aligned}|i\rangle + |-i\rangle &= \sqrt{2}|0\rangle \Rightarrow |0\rangle = \frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle) \\ |i\rangle - |-i\rangle &= i\sqrt{2}|1\rangle \Rightarrow |1\rangle = \frac{-i}{\sqrt{2}}(|i\rangle - |-i\rangle)\end{aligned}$$

$$\begin{aligned}\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) &= \frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle) - \frac{-i}{\sqrt{2}}(|i\rangle - |-i\rangle)) \\ &= (\frac{1}{2} - \frac{i}{2})|i\rangle + (\frac{1}{2} + \frac{i}{2})|-i\rangle\end{aligned}$$

The probability of this outcome for $|i\rangle$:

$$|\frac{1}{2} - \frac{i}{2}|^2 = \frac{1}{2}$$

The probability of this outcome for $|-i\rangle$:

$$|\frac{1}{2} + \frac{i}{2}|^2 = \frac{1}{2}$$

(c) $-|i\rangle$ and $\{|0\rangle, |1\rangle\}$.

$$\begin{aligned} -|i\rangle &= -\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \\ &= \frac{-1}{\sqrt{2}}|0\rangle + \frac{-i}{\sqrt{2}}|1\rangle \end{aligned}$$

The probability of this outcome for $|0\rangle$:

$$|\frac{-1}{\sqrt{2}}|^2 = \frac{1}{2}$$

The probability of this outcome for $|1\rangle$:

$$|\frac{-i}{\sqrt{2}}|^2 = \frac{1}{2}$$

(d) $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and $\{|i\rangle, |-i\rangle\}$.

$$\begin{aligned} |+\rangle + |-\rangle &= \sqrt{2}|0\rangle \Rightarrow |0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \\ |+\rangle - |-\rangle &= \sqrt{2}|1\rangle \Rightarrow |1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle) \end{aligned}$$

$$\begin{aligned} |i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \\ &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) + \frac{i}{\sqrt{2}}(|+\rangle - |-\rangle)\right) \\ &= \frac{1+i}{2}|+\rangle + \frac{1-i}{2}|-\rangle \end{aligned}$$

$$\begin{aligned} |-i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \\ &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) - \frac{i}{\sqrt{2}}(|+\rangle - |-\rangle)\right) \\ &= \frac{1-i}{2}|+\rangle + \frac{1+i}{2}|-\rangle \end{aligned}$$

$$\begin{aligned}
\frac{1}{\sqrt{2}}(|i\rangle - |-i\rangle) &= \frac{1}{\sqrt{2}}\left(\frac{1-i}{2}|+\rangle + \frac{1+i}{2}|-\rangle - \frac{1+i}{2}|+\rangle + \frac{1-i}{2}|-\rangle\right) \\
&= \frac{1}{2\sqrt{2}}(2i|+\rangle - 2i|-\rangle) \\
&= \frac{i}{\sqrt{2}}|+\rangle - \frac{i}{\sqrt{2}}|-\rangle
\end{aligned}$$

The probability of this outcome for $|i\rangle$:

$$\left|\frac{i}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

The probability of this outcome for $|-i\rangle$:

$$\left|\frac{-i}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

(e) $\frac{1}{\sqrt{2}}(|i\rangle - |-i\rangle)$ and $\{|+\rangle, |-\rangle\}$.

$$\begin{aligned}
|i\rangle &= \frac{1+i}{2}|+\rangle + \frac{1-i}{2}|-\rangle && \text{(as proven in part (d))} \\
|-i\rangle &= \frac{1-i}{2}|+\rangle + \frac{1+i}{2}|-\rangle && \text{(as proven in part (d))}
\end{aligned}$$

$$\begin{aligned}
\frac{1}{\sqrt{2}}(|i\rangle - |-i\rangle) &= \frac{1}{\sqrt{2}}\left(\left(\frac{1+i}{2}|+\rangle + \frac{1-i}{2}|-\rangle\right) - \left(\frac{1-i}{2}|+\rangle + \frac{1+i}{2}|-\rangle\right)\right) \\
&= \frac{i}{\sqrt{2}}|+\rangle + \frac{-i}{\sqrt{2}}|-\rangle
\end{aligned}$$

The probability of this outcome for $|+\rangle$:

$$\left|\frac{i}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

The probability of this outcome for $|-\rangle$:

$$\left|\frac{-i}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

(f) $\frac{\sqrt{3}}{2}|+\rangle - \frac{1}{2}|-\rangle$ and $\{|i\rangle, |-i\rangle\}$.

$$\begin{aligned}
|0\rangle &= \frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle) && \text{(as proven in part (b))} \\
|1\rangle &= \frac{-i}{\sqrt{2}}(|i\rangle - |-i\rangle) && \text{(as proven in part (b))}
\end{aligned}$$

$$\begin{aligned}
|+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
&= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle) + \frac{-i}{\sqrt{2}}(|i\rangle - |-i\rangle)\right) \\
&= \frac{1-i}{2}|i\rangle + \frac{1+i}{2}|-i\rangle
\end{aligned}$$

$$\begin{aligned}
|-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
&= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle) - \frac{-i}{\sqrt{2}}(|i\rangle - |-i\rangle)\right) \\
&= \frac{1+i}{2}|i\rangle + \frac{1-i}{2}|-i\rangle
\end{aligned}$$

$$\begin{aligned}
\frac{\sqrt{3}}{2}|+\rangle - \frac{1}{2}|-\rangle &= \frac{3}{\sqrt{2}}\left(\frac{1-i}{2}|i\rangle + \frac{1+i}{2}|-i\rangle\right) - \frac{1}{2}\left(\frac{1+i}{2}|i\rangle + \frac{1-i}{2}|-i\rangle\right) \\
&= \frac{\sqrt{3}-i\sqrt{3}-1-i}{4}|i\rangle + \frac{\sqrt{3}+i\sqrt{3}-1+i}{4}|-i\rangle \\
&= \left(\frac{\sqrt{3}-1}{4} - \frac{\sqrt{3}+1}{4}i\right)|i\rangle + \left(\frac{\sqrt{3}-1}{4} + \frac{\sqrt{3}+1}{4}i\right)|-i\rangle
\end{aligned}$$

The probability of this outcome for $|i\rangle$:

$$\left|\frac{\sqrt{3}-1}{4} - \frac{\sqrt{3}+1}{4}i\right|^2 = \frac{1}{2}$$

The probability of this outcome for $|-i\rangle$:

$$\left|\frac{\sqrt{3}-1}{4} + \frac{\sqrt{3}+1}{4}i\right|^2 = \frac{1}{2}$$

The Artificial Bee Colony (ABC) algorithm

1 Relevant background material

The Artificial Bee Colony (ABC) algorithm is an optimisation algorithm based on the foraging behaviour of the honeybee swarm. It belongs to a class of heuristics that use swarm intelligence, the emergent, 'intelligent', behaviour of a swarm, a collection of (simple) individual agents, to optimise functions. There have been several attempts to model the behaviour of the honeybee swarm.

A Bee Colony Optimisation (BCO) metaheuristic [5] capable of solving combinatorial problems was the first bee-inspired work. This approach was further refined into Bee Swarm Optimisation (BSO) [8], which was adapted to solve the maximum weighted unsatisfiability problem. Other such works include a metaheuristic to solve a 3-SAT problem based on bees' reproduction strategies [6] and a novel routing algorithm called BeeHive [7]. More relevant to ABC is a bee colony system in which simple insectoid robots were used to solve tasks [4]. This paper established the first model of forage selection: interactions between food sources, employed foragers, and unemployed foragers. It also identified two behaviours: the recruitment of an unemployed bee to a food source, and the abandonment of inferior food sources.

However, these works all focus on *combinatorial* problems. The first optimisation algorithm, the Virtual Bee Algorithm (VBA) [9], used the interactions of a swarm of virtual bees to optimise numeric functions. The paper discussed in this work builds upon previous work by the authors in which the ABC algorithm is first described [10] and is compared to a genetic algorithm (GA) [11].

2 Natural language description

In the ABC algorithm, a colony consists of three types of bee:

- Employee: a bee going to a food source that it previously visited.
- Onlooker: a bee waiting in the dance area to choose a food source.
- Scout: a bee carrying out a random search for a new food source.

A *food source* is defined by a position in the state space, a possible solution to the optimisation problem, and a *nectar amount*, the fitness of that solution. There are three control parameters in ABC: the number of food sources, S_n ; the number of iterations a population undergoes, C_{max} ; and *limit*, the number of times the area around a food source is sampled before, if no better food source is discovered nearby, it is discarded.

There are four phases. The initialisation phase is performed once. The employed, onlooker, and scout bee phases are repeated until C_{max} has been reached, or until the nectar amount of a food source exceeds some threshold.

1. **Initialisation.** A randomly distributed initial population (of size S_n) of food sources is generated. Each food source position is a D -dimensional vector, where D is the number of optimisation parameters. The nectar amounts of each food source are computed, and the number of trials for each food source is set to 0.
2. **Employed phase.** Each employed bee goes to the food source in her memory. She produces a probabilistic modification of her food source position using 2.1 and evaluates the nectar amount of the modified position. If it is greater than her original food source, she updates her stored food source. Otherwise, she discards the new position. If the new position is discarded, a counter tracking the number of times the area around her food source has been searched is incremented. She returns to the hive and shares the nectar amount of her food source to the onlooker bees.
3. **Onlooker phase.** Each onlooker bee observes the dance of each employed bee. The onlooker chooses a food source using 2.2. She goes to that source and follows the same procedure as an employed bee: she produces a modification of her food source position, evaluates the nectar amount, updates her food source if it is superior, and discards it otherwise. She also updates the counter.

4. **Scout bee phase.** The food source that has been sampled the most is examined. If it has been sampled more than *limit*, a scout bee find a new food source and updates the value of the old food source.

2.1 Modifying a food source

A food source x is modified into a new food source v using $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$, where $k \in \{1, 2, \dots, S_n\}$ and $j \in \{1, 2, \dots, D\}$ are randomly chosen indices. ϕ_{ij} is a random number in $[-1, 1]$. It controls the production of food source positions near x_{ij} . As $|x_{ij} - x_{kj}|$ decreases, so too does the perturbation of x_{ij} .

2.2 Selecting a food source

A food source is selected using roulette-wheel selection (stochastic sampling with replacement). The probability of a food source i being selected is $p_i = \frac{f(i)}{\sum_{n=1}^{S_n} f(i)}$, where $f(i)$ returns the nectar amount of i .

3 Experiment details

3.1 Comparator algorithms

Three population-based stochastic optimisation algorithms were compared to ABC:

- **Particle Swarm Optimisation (PSO):** uses swarm intelligence. A swarm of particles moves around a search space according to formulae over their position and velocity and the best known positions in the search space. PSO is governed by ω , the inertial weight of a particle, two learning factors ϕ_1 and ϕ_2 , and the formulae relating to acceleration, velocity, and position.
- **Genetic Algorithms (GA):** belong to a class of algorithms called Evolutionary Algorithms (EA) which simulate evolution. Solutions are encoded as the genes of individuals, which mutate, breed, and die. GAs are governed by the probability of crossover P_c , the likelihood of two individuals being combined, the probability of mutation P_m , the likelihood of a gene changing, and the *operators* selected for mutation, crossover, and selection.
- **Particle Swarm Evolutionary Algorithms (PS-EA):** a hybrid of PSO and EA in which particles are adaptively updated depending on their fitness. This algorithm seeks to dynamically adjust the inheritance probabilities based on the convergence rate of an algorithm in a particular iteration.

3.2 Benchmark functions

A function is *multimodal* if it has multiple optima and *multivariable* if it has multiple input parameters. Five classic benchmark functions [12] were used to test the performance of ABC against PSO, PS-EA, and GA:

- **Griewank function:** has a product term that introduces interdependence among the variables, penalising techniques which optimise each variable independently.
- **Rastrigin function:** 0 at its global minimum at the origin. Produces many local, regularly distributed minima, so an optimisation algorithm can easily be trapped.
- **Rosenbrock function:** has a global optimum in a long, narrow, parabolic, flat valley. It is easy to find a local optimum, but difficult to converge. Variables are therefore strongly dependent. Gradients generally do not point towards the optimum.
- **Ackley function:** has an exponential term that covers its surface with numerous local optima. It tests how efficiently an algorithm both explores and exploits.
- **Schwefel function:** has a surface comprised of a great number of peaks and valleys, with the second-best minimum far from the global minimum, which itself is near the bounds of the domain.

3.3 Test parameters

Common control parameters of the algorithms are population size (S_n in ABC) and the number of generations (C_{max}). In the experiments, a C_{max} of 500, 750, and 1000 was used with the dimensions 10, 20, and 30, respectively. Population size was held constant at 125. A second run of ABC used 1000, 1500, and 2000 cycles with dimensions 10, 20, and 30 respectively.

ABC split bees evenly into 50% onlooker and 50% employed, with a single scout. Each experiment was repeated 30 times with different random seeds.

The GA used single-point uniform crossover (swapping one gene at the same point in two individuals) with a probability $P_c = 0.95$. The paper states that individuals were selected randomly, but this would defeat the whole purpose of a GA, so this is deemed to be incorrect. Individuals were ranked with a linear fitness function. Gaussian mutation occurred with probability $P_m = 0.1$.

The PSO used a inertia weight ω which varied linearly from 0.9 to 0.7 with the number of iterations. The learning factors ϕ_1 and ϕ_2 were both set to 2. The lower and upper velocity bounds (V_{min} and V_{max}) of the particles are set to the lower and upper of x (x_{min} and x_{max}).

4 Results overview

The paper found that the ABC algorithm performs better than PS-EA on the Greiwank and Ackley functions. It is only outperformed on the Schwefel function by PS-EA and GA for dimensions 20 and 30. After increasing the C_{max} , the ABC algorithm converged to the minimum of the Schwefel function as well. Increasing C_{max} considerably improved the performance of ABC, though the performance of PSO and PS-EA remained stagnant, suggesting premature convergence. It is successful in optimising multivariable multimodal functions.

5 Detailed pseudocode description

See overleaf.

Algorithm 1 ABC

```
1: procedure ABC
2:    $S_n$ : number of food sources
3:    $limit$ : number of trials before abandoning a food source
4:    $C_{max}$ : number of cycles
5:    $f(x)$ : function returning the fitness of a solution  $x$ 
6:    $n()$ : function returning a random solution
7:    $m(x)$ : function returning a modification of  $x$  as per 2.1
8:    $p(x)$ : function returning the probability of selecting a solution  $x$  as per 2.2
9:   Initialisation:
10:  for  $s \leftarrow 1$  to  $S_N$  do
11:     $X(s) \leftarrow n()$ 
12:     $f_s \leftarrow f(X(s))$ 
13:     $trial(s) \leftarrow 0$ 
14:  end for
15:  Loop:
16:     $c \leftarrow 0$ 
17:    while  $c < C_{max}$  do
18:      //Employed bee phase
19:      for  $s = 1$  to  $S_N$  do
20:         $x' \leftarrow m(X(s))$ 
21:        if  $f(x') < f_s$  then
22:           $X(s) \leftarrow x'$ 
23:           $f_s \leftarrow f(x')$ 
24:           $trial(s) \leftarrow 0$ 
25:        else
26:           $trial(s) \leftarrow trial(s) + 1$ 
27:        end if
28:      end for
29:      //Onlooker bee phase
30:       $s \leftarrow 1$ 
31:       $t \leftarrow 1$ 
32:      while  $t < SN$  do
33:         $r \leftarrow rand(0, 1)$ 
34:        if  $r < p(s)$  then
35:           $t \leftarrow t + 1$ 
36:           $x' \leftarrow n()$ 
37:          if  $f(x') < f_s$  then
38:             $X(s) \leftarrow x'$ 
39:             $f_s \leftarrow f(x')$ 
40:             $trial(s) \leftarrow 0$ 
41:          end if
42:        end if
43:      end while
44:      //Scout bee phase
45:       $m \leftarrow \{s : trial(s) = max(trial)\}$ 
46:      if  $trial(m) \geq limit$  then
47:         $X(m) \leftarrow n()$ 
48:         $f_m \leftarrow f(X(m))$ 
49:         $trial(m) \leftarrow 0$ 
50:      end if
51:      Store the best solution
52:       $c \leftarrow c + 1$ 
53:    end while
54:    Return the best solution
55: end procedure
```

References

- [1] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J. Glob. Optim.* 39 (2007) 459–471.
- [2] D. Karaboga and B. Gorkemli, A combinatorial artificial bee colony algorithm for Travelling Salesman problem, *Proc. of Int. Symp. on Innovations in Intelligent Systems and Applications (INISTA)*, IEEE Press (2011) 50–53.
- [3] X. Zhang, Q. Bai and X. Yun, A new hybrid artificial bee colony algorithm for the Traveling Salesman Problem, *Proc. of 3rd Int. Conf. on Communication Software and Networks (ICCSN)*, IEEE Press (2011) 155–159.
- [4] V. Tereshko and A. Loengarov, Collection Decision-Making In Honey Bee Foraging Dynamics, *Comput. Inf. Sys J.* (2005) 9(3), 1-7
- [5] D. Teodorovic, Transport Modeling By Multi-Agent Systems: A Swarm Intelligence Approach, *Transport. Plan. Technol.* (2003) 26(4), 289-312
- [6] K. Benatchba, L. Admane, M. Koudil, Using bees to solve a data-mining problem expressed as a max-sat one, artificial intelligence and knowledge engineering applications: a bioinspired approach, *Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC* (2005)
- [7] H.F. Wedde, M. Farooq, Y. Zhang, BeeHive: an efficient fault-tolerant routing algorithm inspired by honeybee behaviour, ant colony, optimisation, and swarm intelligence, *Proceedings of the 4th International Workshop, ANTS* (2004)
- [8] H. Drias, S. Sadeg, S. Yahli, Cooperative bees swarm for solving the maximum weighted satisfiability problem, computational intelligence and bioinspired Systems, *Proceedings of the 8th International Workshop on Artificial Neural Networks, IWANN* (2005)
- [9] X. S. Yang, Engineering optimizations via nature-inspired virtual bee algorithms, *Lecture Notes in Computer Science*, pp. 317-323. Springer, GmbH (2005)
- [10] D. Karaboga, An idea based on honey bee swarm for numerical optimization, *Technical Report-TR06*, Erciyes University (2005)
- [11] B. Basturk and D. Karaboga, An artificial bee colony (ABC) algorithm for numeric function optimization, *Proceedings of the IEE Swarm Intelligence Symposium* (2006)
- [12] D. Srinivasan and T. H. Seow, Evolutionary Computation, *CEC* (2003)