University of Maribor

Faculty of Electrical Engineering and Computer Science

A Hybrid Artificial Bee Colony Algorithm for Graph 3-Coloring

ICAISC - Symposium on Swarm Intelligence & Differential Evolution SIDE 2012

Iztok Fister Jr.

Iztok Fister

Janez Brest

Zakopane, 29. April – 2. May 2012

Contents

Motivation

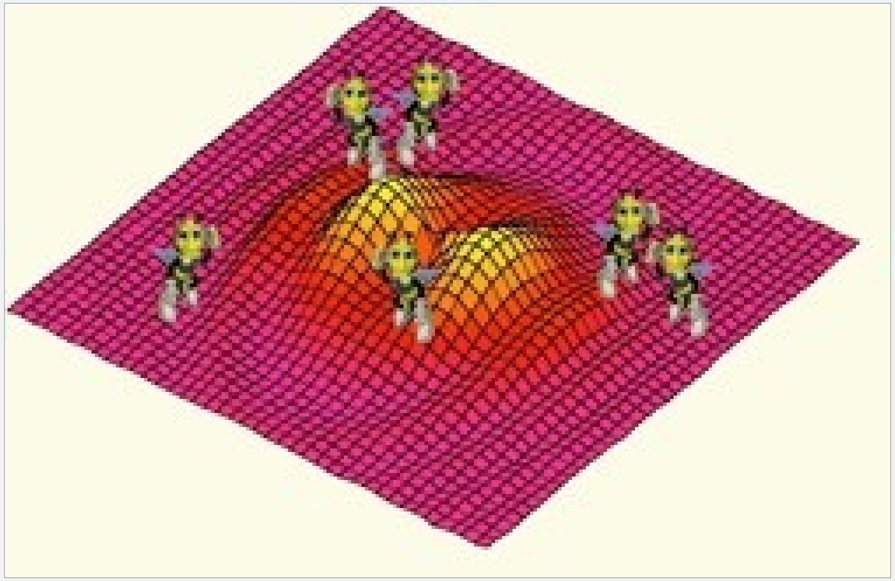
- To show that ABC algorithms can also be used by solving of combinatorial optimization problems.
- To show that the HABC is comparable with the well-known heuristics for graph coloring, as Tabucol (Hertz & de Werra, 1987) and HEA (Galinier & Hao, 1999).
- To show that HABC improves the results of the EA-SAW (Eiben et al. 1998) that is well-known evolutionary algorithm for graph 3-coloring.

Characteristics of the problem

- NP-complete problem
- Constraint Optimization Problem (COP)
- The hardest instances occur near to the *phase transition* (Turner, 1988; Petford & Welsh, 1989; Hayes, 2003)
- Existing algorithms:
  - DSatur (Brelaz, 1979) greedy algorithm,
  - Tabucol (Hertz & de Werra, 1986),
  - EA with method SAW (Eiben et al. 1998),
  - HEA (Galinier & Hao, 1999).

Artificial Bee Colony Algorithm

- An Artificial Bee Colony (ABC) was proposed by Karaboga in 2005.
- ABC is based on a particular intelligent behavior of honeybee swarms (Swarm Intelligence).



Agents in ABC:

- The Employed Bee,
- The Onlooker Bee,
- The Scout.

Elements of the MABC algorithm

- **Representation**: real-valued vector $Y_i = \{w_{ij}\}$ representing food sources
- **Initialization**: random position of food sources
- **Move employed bees**: looking for a better food sources
- **Move onlooker bees**: moving towards the more promising food sources
- **Move scouts**: employed bees on food sources that does not improve become scouts
- **Fitness calculation**: evaluate the nectar amounts of the particular food source
- **Terminate condition**: number of function evaluations

Moving the employed bees

- Calculation of the new position:

$$y_{ij}(t+1) = y_{ij}(t) + \phi_{ij}\left(y_{ij}(t) - y_{kj}(t)\right)$$

  - y_i: The position of the employed bee.
  - t : The iteration number.
  - y_k : The randomly chosen employed bee.
  - j : The dimension of the solution.
  - ϕ_{ij} : The random value in the range $\lfloor -1, 1 \rfloor$.
- Stochastic large distance exploration

Moving the onlooker bees

- Moving to the *i*-th food source with probability:

$$p_i = \frac{f(s_i)}{\sum_{k=1}^{NP} f(s_k)}$$

  - p_i : The probability of selecting the *i*-th employed bee as the onlooker.
  - *NP* : The number of bees into hive.
  - s_i : The *i*-th solution.
  - $f(y_i)$: The fitness value of the *i*-th solution.
- Stochastic moderate-distance exploration

Moving the scouts

- The food sources that cannot be improved within limit number of cycles are changed as:

$$y_i(t+1) = y_i(t) + \lambda . U_i$$

  - y_i : The position of the employed bee.
  - t : The iteration number.
  - λ : The prescribed step size length.
  - U_i : The unit random vector generated for the i-th solution.
- Deterministic short-distance exploration (local search)

Fitness calculation

- Food source represents weigths determining the order in which vertices are to be colored
- Fitness calculation consists of two steps:
  - Weights are transformed into a vertex permutation,
  - The permutation is decoded into graph 3-coloring by DSatur construction heuristic.
- Fitness function counts the number of uncolored vertices

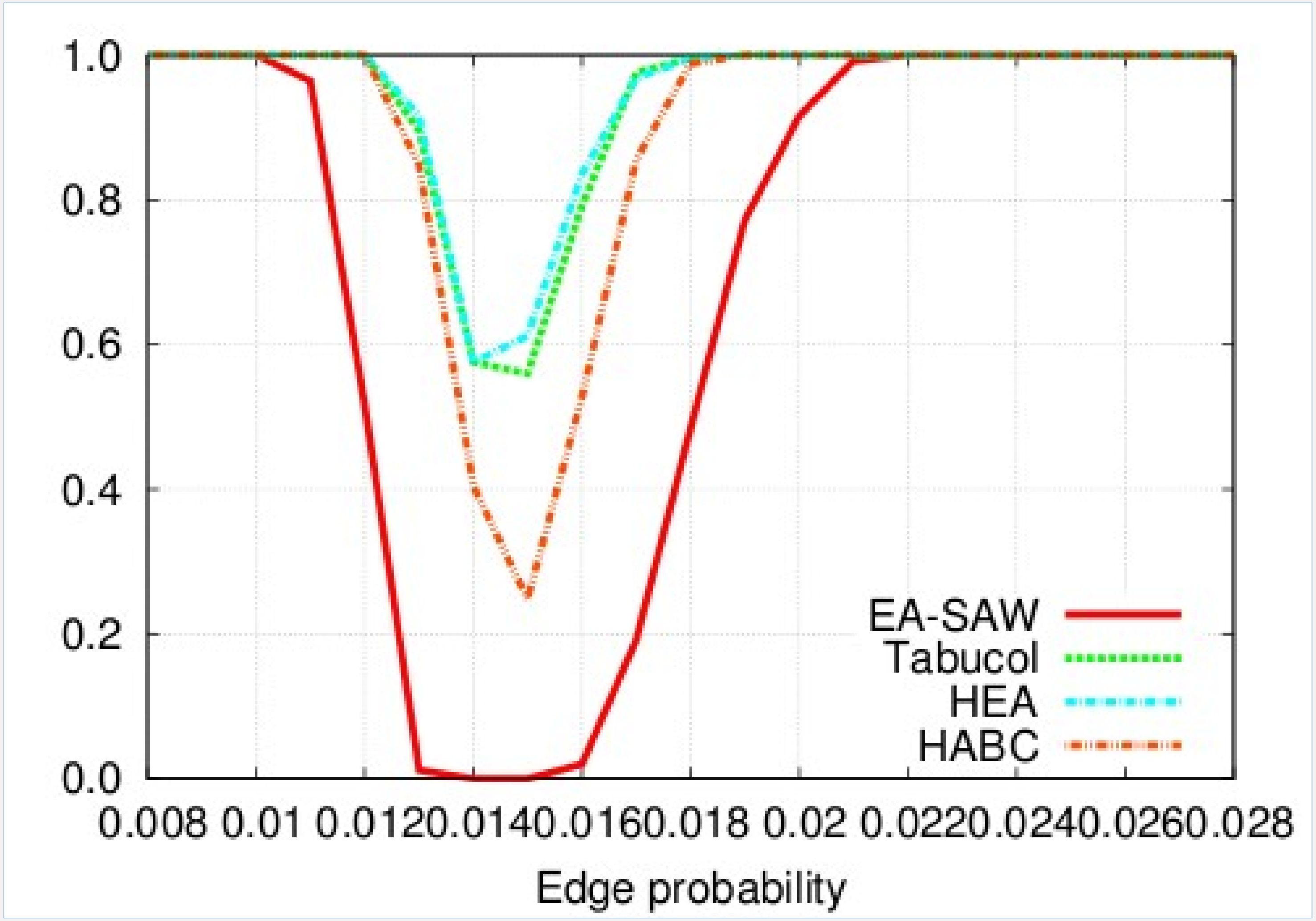
Experiments and results

- Comparison with:
  - Tabucol (Hertz & de Werra, 1986),
  - EA with method SAW (Eiben & al. 1998),
  - HEA (Galinier & Hao, 1999).
- All implementation can be downloaded from Internet
- Characteristics of HABC:
  - Population size: *NP*=100,
  - Number of objective function evaluations: *FEs*=300,000,
  - Number of runs: 25,
  - Scout *limit*: 1,000.
- Measures:
  - Success rate (*SR*)
  - Average number of function evaluation (*AES*).

Test problems

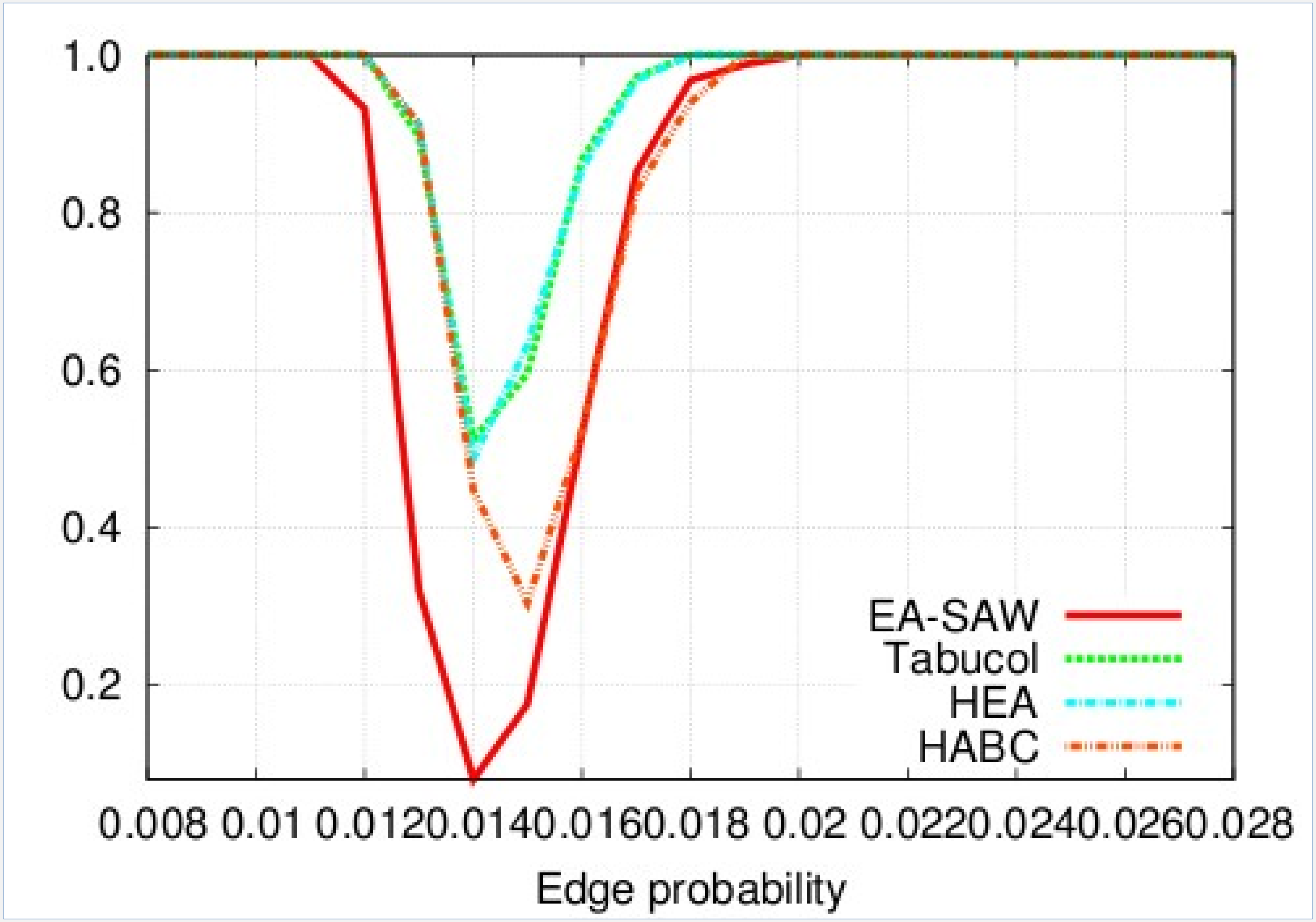
- Random graphs generated by Culbersone graph generator with following characteristics:
  - *types*: equi-partite, uniform, flat;
  - *sizes*: n=500 (also medium-scale graphs);
  - *edge density*: p=0.008…0.028 in steps of 0.001;
  - *seed*: s=1…10.
- Capture the phenomenom of *phase transition*, where graphs are really hard to color
- Phase transition occurs when p=0.014

Influence of the edge density 1/3



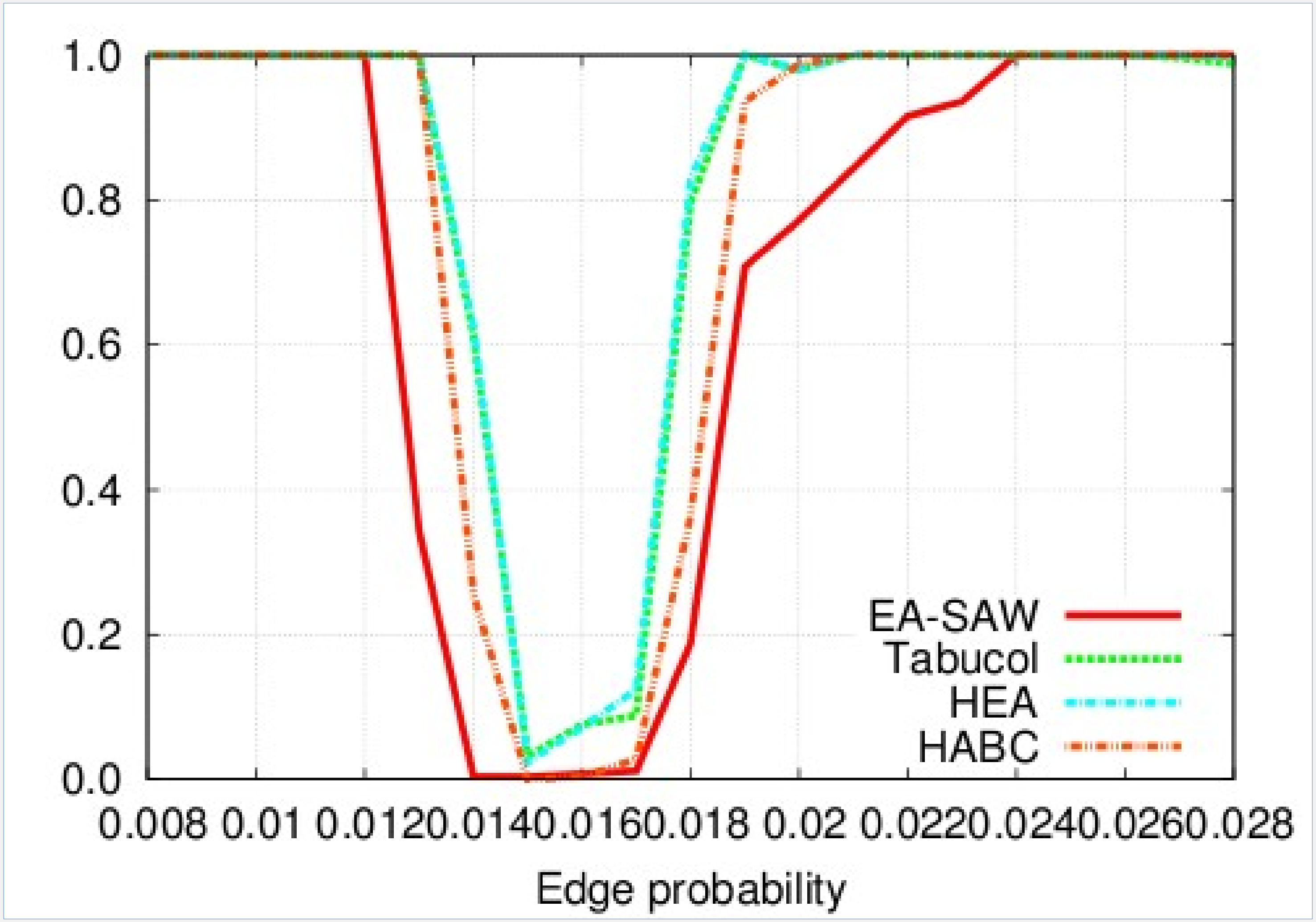
- SR by uniform graphs.

Influence of the edge density 2/3



- SR by equi-partite graphs.

Influence of the edge density 3/3



- SR by flat graphs.

Influence of the local search

p	Uniform		Equi-partite	
	Rand.	RWDE	Rand.	RWDE
0.013	0.816	0.848	0.872	0.912
0.014	0.112	0.404	0.200	0.448
0.015	0.060	0.248	0.036	0.304
0.016	0.180	0.528	0.104	0.524
0.017	0.328	0.856	0.340	0.828
average	0.299	0.577	0.310	0.603

Conclusion

- The results of HABC is comparable with the results of Tabucol and HEA on medium-scale graphs
- HABC improves results of EA-SAW by solving of all three type of medium-scale graphs
- Therefore, HABC could be successfully applied to solving of graph 3-coloring problem as well
- In the future, tests on large-scale graphs (1,000 vertices) would be conducted