Hybridization of Estimation of Distribution Algorithms with a Repair Method for Solving Constraint Satisfaction Problems

Hisashi Handa

Okayama University, Tsushima-Naka 3-1-1,
Okayama 700-8530, JAPAN,
handa@sdc.it.okayama-u.ac.jp,
http://www.sdc.it.okayama-u.ac.jp/~handa/index-e.html

Abstract. Estimation of Distribution Algorithms (EDAs) are new promising methods in the field of genetic and evolutionary algorithms. In the case of conventional Genetic and Evolutionary Algorithm studies to apply Constraint Satisfaction Problems (CSPs), it is well-known that the incorporation of the domain knowledge in the CSPs is quite effective. In this paper, we propose a hybridization method (memetic algorithm) of Estimation of Distribution Algorithms with a repair method. Experimental results on general CSPs tell us the effectiveness of the proposed method.

1 Introduction

As the scale and the complexity of engineering problems increases, the distributed processing approaches inspired by the constraints-oriented problem solving methods are now receiving attentions. The notion of Constraint Satisfaction Problems (CSPs) provides us general framework adaptable to a wide variety of problems in various fields [1,2]. The CSPs are a problem class which consists of variables and constraints on the variables. Solving problems by using Constraint-oriented approach is that, first, we have to describe constraints which should satisfy between elements in the target environments. Then, we employ "constraint satisfaction problem solver (CSP solver)," to find satisfied solutions of the described problem instance such that all constraints in the problem are satisfied. Recently, CSP solvers from genetic and evolutionary algorithms have been broadly studied by many researchers [2]-[9]. These studies showed that the uses of the domain knowledge in the CSPs, that is, the hybridization with local search method (repair method) based upon the notion of Min-Conflict Hill Climbing (MCHC) [10], and the utilization of constraint networks [6], are quite effective.

Estimation of Distribution Algorithms (EDAs) are new promising methods in the field of genetic and evolutionary algorithms [11,12]. The EDAs employ the probabilistic model, which is constituted by a database containing the genetic information of the selected individuals in the previous generation, to yield a

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 991–1002, 2003.

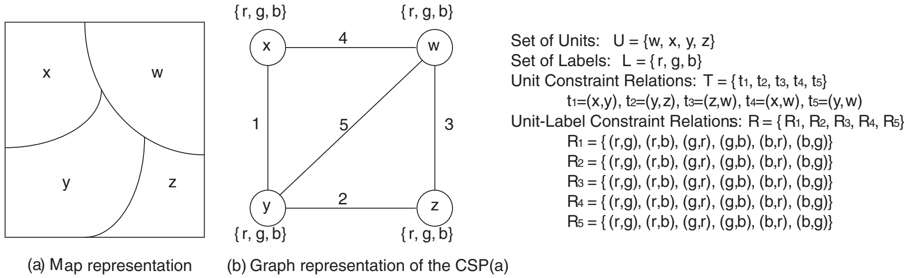

Fig. 1. An example of CSP: graph coloring problem

new population. Hence, in the EDAs, Genetic operations such like crossover and mutation are not adopted.

In this paper, we propose a new evolutionary constraints satisfaction problem solver incorporating a repair method into the EDA. Moreover, we also introduce a manner which incorporates the knowledge of the constraint network to the Bayesian Network structure search.

Related works are described as follows: Tsang wrote comprehensive text book about CSPs which is also written about genetic approach for solving CSPs [1]. Eiben summarized how to solve several classes of CSPs by using GAs in [3,4]. Riff proposed a fitness function and genetic operators for solving CSPs effectively, which are utilizing the knowledge with regard to the constraint network [5,6]. Coevolutionary Evolutionary Computations were often adopted to solve CSPs [2,9]. With respect to the EDAs, Larrañaga and Lozano edited a comprehensive book of the EDAs[12]. EDAs in section 3 are introduced by referring to this textbook. Genetic algorithms with local search methods are often called "memetic algorithms," and have been studied by many researcher for the last decade [13].

In the next section, the basics of the CSPS will be described. Then we will briefly introduce three kinds of EDAs in section 3, which are employed for our experiments. Section 4 will introduce the proposed method. Then, experiments on general Constraint Satisfaction Problems will be carried out for conventional GEA approach, EDAs introduced in section 3 and the proposed method. Section 6 will conclude this paper.

2 Constraint Satisfaction Problems

2.1 Formulation

Constraint Satisfaction Problems (CSPs) are a class of problems which consists of variables and constraints on the variables [1]. In addition, the class of CSPs in which each constraint in the problems is related only to two variables is called binary CSPs. In this paper, we treat a class of discrete binary CSPs, where the

```
Procedure Min-Conflict Hill Climbing
begin
  e(i) ← evaluate each variable i in the current solution
  until Stopping criteria is hold
    i* ← Select a variable with the worst evaluation
    Re-evaluate e(i*) for all labels in the variable i*
    Select a label in the variable i* with the best evaluation
    Modify the current solution to the selected label
    Re-evaluate the current solution with respect to the modification
  end
end
```

Fig. 2. Pseudo code of Min-Conflict Hill Climbing

word 'discrete' means that each variable is associated with a finite set of discrete values (labels) that are candidate values of the variable. An example of the graph coloring problem [10], a binary CSP which is one of the benchmark CSP is delineated in Fig. 1. As depicted in the figure, CSPs are defined by (U, L, T, R). U, L, T and R denote a set of units (variables), a set of labels (values), unit constraint relations and unit-label constraint relations, respectively.

In this paper, we use two indices, tightness and density to analyse the difficulty of CSPs [1]. The tightness of an edge ij is given as the ratio of the number of satisfying 2-compound labels (in unit-label constraint relations) on the edge ij over the number of all 2-compound labels on the edge ij. Furthermore, the tightness of a problem is given by the averaged value of tightness of the edges in the problem. The density of a problem indicates the proportion of constraint relations that actually exist between any pair of nodes. Furthermore, the number of 2-compound labels on the edge XY is the same as the product of the number of labels on each nodes, that is, 3×3.

2.2 Min-Conflict Hill Climbing

This local search method, often called heuristic repair method, is adopted not only from genetic and evolutionary algorithms but also from approximation algorithms for solving CSPs. The procedure of the MCHC is described in Fig. 2: The MCHC begins with a given solution. First, the solution is evaluated. In that time, the number of constraint violations for each variable is memorized. Then, the variable which have the most constraint violations, i.e., the least evaluation, is chosen. If some variables tie with each other, the variable is randomly chosen among them. For the selected variable, all the labels are examined and evaluated. In similar to the variable selection, one of labels with the least constraint violations is chosen (randomly in the case of tie). Re-evaluation for above modification is carried out, and the process returns to the variable selection phase.

Procedure Estimation of Distribution Algorithm
begin
  initialize D_0
  evaluate D_0
  until *Stopping criteria is hold*
    $D_{l-1}^{Se} \leftarrow$ Select N individuals from D_{l-1}
    $p_l(\mathbf{x}) \leftarrow$ Estimate the probabilistic model from D_{l-1}^{Se}
    $D_l^{Se} \leftarrow$ Sampling M individuals from $p_l(\mathbf{x})$
  end
end

Fig. 3. Pseudo code of Estimation of Distribution Algorithms

3 Brief Introduction of the Estimation of Distribution Algorithms

3.1 General Framework of EDAs

The Estimation of Distribution Algorithms are a class of evolutionary algorithms which adopt probabilistic models to reproduce the genetic information of the next generation, instead of conventional crossover and mutation operations. The probabilistic model is represented by conditional probability distributions for each variable (locus). This probabilistic model is estimated from the genetic information of selected individuals in the current generation. Hence, the pseudo-code of EDAs can be written as Fig. 3, where D_l, D_{l-1}^{Se}, and $p_l(\mathbf{x})$ indicate the set of individuals at lth generation, the set of selected individuals at $l-1$th generation, and estimated probabilistic model at lth generation, respectively [12]. The representation and estimation methods of the probabilistic model are devised by each algorithm. The following subsections will overview some EDAs. For a more thorough overview, see [12,14].

3.2 UMDA

UMDA (Univariate Marginal Distribution Algorithm) was introduced by Mühlenbein [12]. As indicated by its name, the variables of the probabilistic model in this algorithm is assumed to be independent from other variables. That is, the probability distribution $p_l(\mathbf{x})$ is denoted by a product of univariate marginal distributions, i.e.,

$$p_l(\mathbf{x}) = \prod_{i=1}^{n} p_l(x_i),$$

where $p_l(x_i)$ denotes the univariate marginal distribution $X_i = x_i$ at a variable X_i at generation l. This univariate marginal distribution is estimated from marginal frequencies:

$$p_l(x_i) = \frac{\text{the number of solutions, where } X_i = x_i \text{ in the selected individuals}}{N},$$

where N denotes the number of selected individuals, which is fixed in advance.

3.3 MIMIC

De Bonet *et al.* proposed MIMIC [12,17], a kind of EDAs whose probabilistic model is constructed with bivariate dependency such like COMIT [18]. While the COMIT generates a tree as dependency graph, the probabilistic model of the MIMIC is based upon a permutation π.

$$p_l(\mathbf{x}) = \prod_{j=1}^{n-1} p_l(x_{i_{n-j}}|x_{i_{n-j+1}}) \cdot p_l(x_{i_n}),$$

where the permutation π is represented by $(i_1, i_2, \ldots, i_n)$. The permutation π is decided in each generation such that the following Kullback-Leibler divergence $H_l^\pi(\mathbf{x})$ is minimized:

$$H_l^\pi(\mathbf{x}) = h_l(X_{i_n}) + \sum_{j=1}^{n-1} h_l(X_{i_j}|Y_{i_{j+1}}),$$

where $h_l(X) = -\sum_x p(X=x)\log p(X=x)$, and $h_l(X|Y) = -\sum_x \sum_y p(Y=y)p(X=x|Y=y)\log p(X=x|Y=y)$. However, such minimization is NP so that this minimization is carried out by greedy search.

3.4 EBNA

Like BOA and LFDA [15,16], The EBNA (Estimation of Bayesian Networks Algorithms) adopts Bayesian Network (BN) as the probabilistic model, which is proposed by Larrañaga *et al.* [12,19]. They proposed several kinds of EBNAs, such as EBNA$_{PC}$, EBNA$_{K2+pen}$, EBNA$_{BIC}$, and so on. Here, we introduce only EBNA$_{BIC}$ used in our experiments. EBNA$_{BIC}$ searches for the better structure of BN by using search+score method. In the case of the EBNA$_{BIC}$, scoring is achieved by penalized maximum likelihood $BIC(S,D)$ for a given structure S and a dataset D, called Bayesian Information Criteria, denoted by the following equation:

$$BIC(S,D) = \sum_{i=i}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} - \frac{1}{2}\log N \sum_{i=1}^{n} q_i(r_i-1),$$

where the structure S is represented by Direct Acyclic Graphs, n is the number of variables of the Bayesian Network, r_i is the number of different values that variable X_i can take, q_i is the number of different values that the parent variables of X_i in the structure S can take, N_{ij} is the number of individuals in D in which the parent variables of variable X_i take their j^{th} value, and N_{ijk} is the number of individuals in D in which variable X_i takes its k^{th} value and the parent variables of the variable i take their j^{th} value [12]. As the search method in the EBNA$_{BIC}$, an arc-based local search is adopted due to the NP property of searching the best structure for BNs.

Procedure EDA for CSP
begin
  initialize D_0
  evaluate D_0
  until *Stopping criteria is hold*
    $D_{l-1}^{Se} \leftarrow$ Select N individuals from D_{l-1}
    $p_l(\mathbf{x}) \leftarrow$ Estimate the probabilistic model from D_{l-1}^{Se}
    $D_l^{\prime Se} \leftarrow$ Sample M individuals from $p_l(\mathbf{x})$
    $D_l^{Se} \leftarrow$ **carry out a Repair Algorithm to** $D_l^{\prime Se}$
  end
end

Fig. 4. Pseudo code of EDA for CSP

4 The Proposed Method

As mentioned in the introduction of this paper, applying conventional GEA to solve CSPs have been studied by many researchers. According to the conclusions of their studies, the utilization of the CSP-specific knowledge, such as the topology of constraint networks, local constraint evaluation (sub-evaluation of individuals), and so on, yields the great improvement of the search ability of GEAs [3]-[9]. Hence, we propose a hybrid method of EDAs with repair method. Moreover, we also introduce a manner which incorporates the knowledge of the constraint network to the BN structure search.

As a repair method for the proposed method, we employ asexual heuristic operator in H-GA proposed by Eiben *et al.* [3]. The asexual heuristic operator uses the notion of the MCHC described in Fig. 2 to operate the genetic information of individuals. The number of applying the MCHC is pre-defined, that is, 1/4 of the string length. The timing to apply this operation to the EDA population is after sampling new individuals from the estimated (joint) probability distribution as denoted in the Fig. 4.

Furthermore, in order to reduce the computational effort, the variable candidate selection guided by constraint networks is introduced to bivariate or multivariate dependent algorithms. That mechanism is quite simple: arcs which do not have constraint-relations, which can be referred by the constraint network, do not calculate its indices to construct the probabilistic model, such like $H_l^{\pi}(\mathbf{x})$ and $BIC(S, D)$. However, it saves a large amount of computational time since the calculation of conditional probabilities takes much time and appears for corresponding candidate structures.

5 Experimental Results

In this paper, we carry out several experiments based on various general CSPs that are generated randomly for a wide variety of "density" and "tightness" of constraint conditions in the CSPs that are the basic measures of characterizing CSPs and are described in section 2. The general CSPs are randomly generated

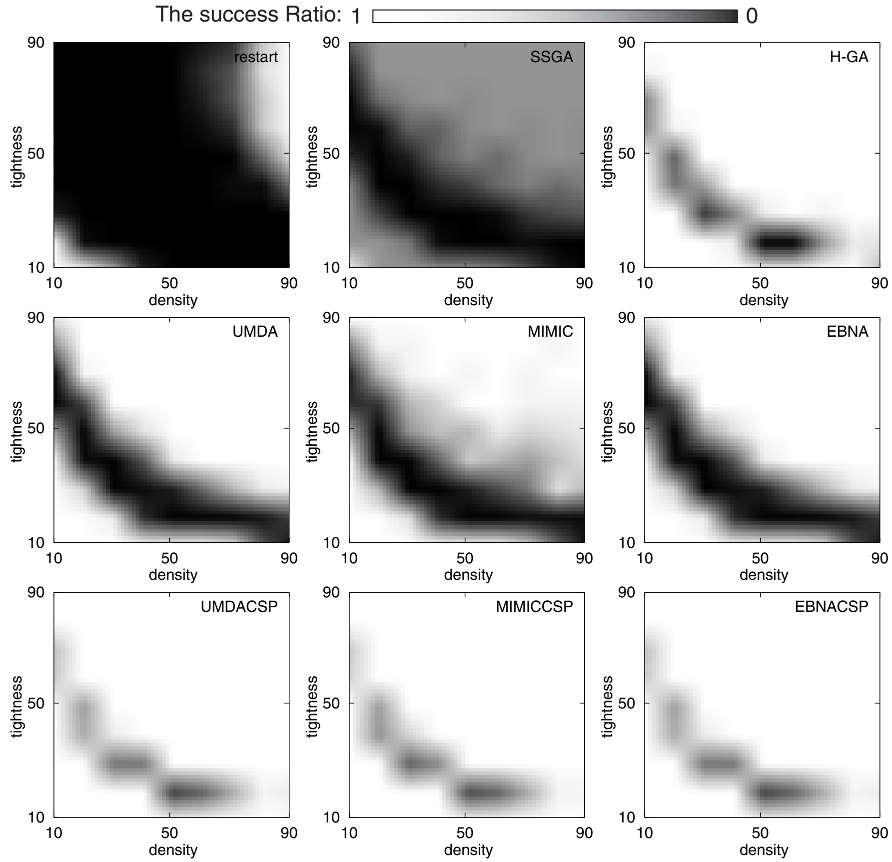

Fig. 5. Experimental results on general CSP, the ratio of success examination: results with higher success ratio tend to be brighter, and results with lower success ratio tend to be darker; The upper row is for conventional methods, restart method with MCHC, Steady-State Genetic Algorithms, and H-GA; The middle row is for EDAs, UMDA, MIMIC, and EBNA; The lower row is for the proposed methods, UMDACSP, MIMICCSP, and EBNACSP.

as follows: First, specify the tightness and density in the sense in section 2. Next, for all combination of two indices, decide whether unit constraint relation is set to each of the pairs of variables by taking account of the value of density. Finally, for all unit constraint relations, the number of the unit-label constraint relationships is set to be directly proportional to the tightness.

In order to solve the general CSPs, i.e., to find a solution such that it has no constraint violation, we introduce following fitness function which is generally adopted:

$$1 - N_{CV}/N_{MC},$$

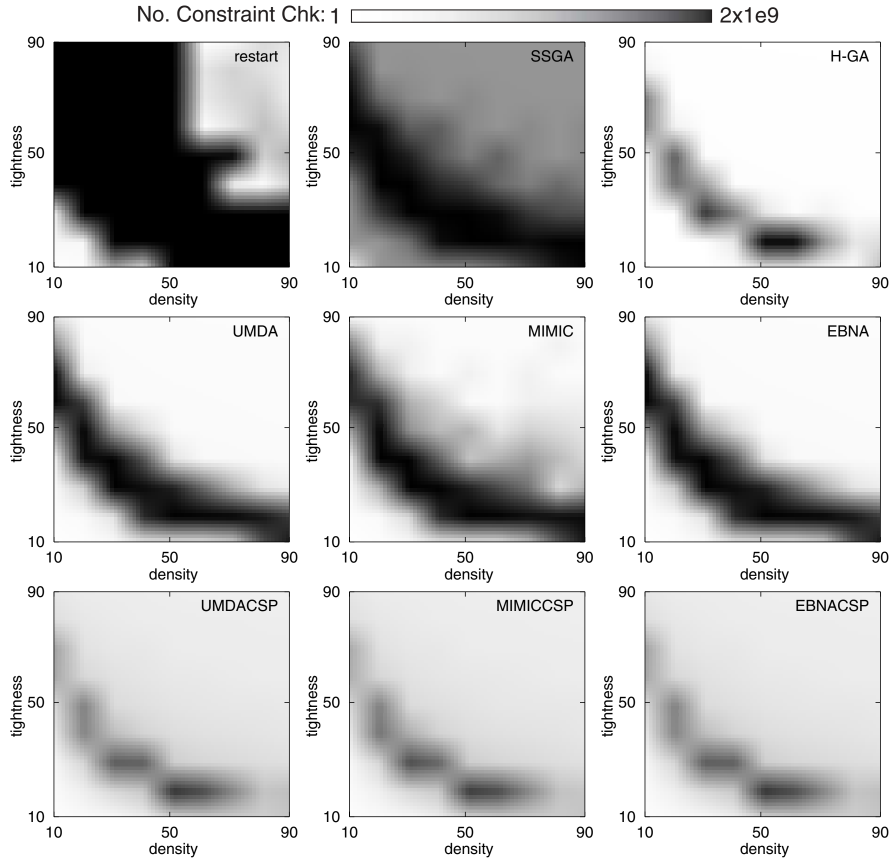

Fig. 6. Experimental results on general CSP, the number of constraint checks until finding satisfiable solutions when satisfiable solutions are found within 2 billion constraint checks: results with faster discovers tend to be brighter, and results with later discovers tend to be darker; The allocation of each graph is the same as the previous figure.

where N_{CV} and N_{MC} indicate the number of constraint violations in a certain individual, and the number of possible constraint relations in a problem with n variables, i.e., $n(n-1)/2$ in the case of binary CSPs, respectively. The coding method adopted in this paper is a naive one such that each variable in a problem instance is corresponding to each gene. That is, each label associated with each variable is directly represented as each allele at corresponding locus.

First, we compare the proposed methods for UMDA, MIMIC, and EBNA, i.e., UMDACSP, MIMICCSP, and EBNACSP, respectively, with various kinds of conventional method: restart method, steady-state GA, H-GA proposed by Eiben et al., UMDA, MIMIC, and EBNA [21]. Parameters for each algorithm are described as follows: The restart method is a (non-evolutionary) conventional

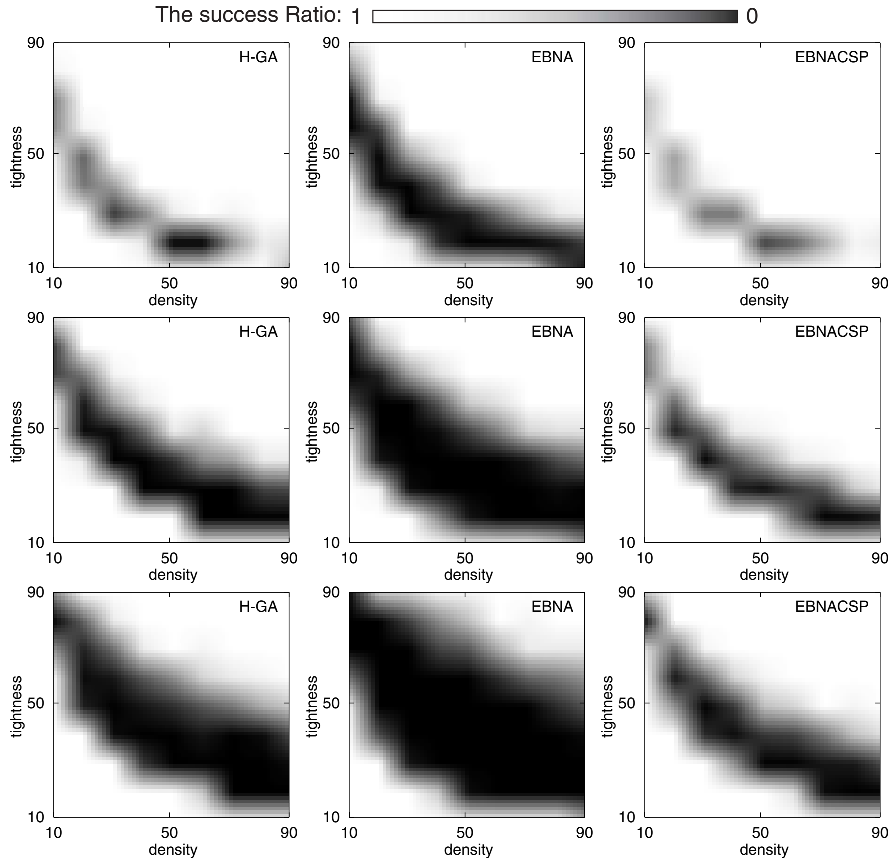

Fig. 7. The scalability of each algorithm, the ratio of success examination: the depiction manner of each graph is the same as Fig. 5; 40 variables with 10(UPPER ROW), 20(MIDDLE ROW), and 30(LOWER ROW) labels in each variable; H-GA(LEFT COL.), EBNA(MIDDLE COL.), and EBNACSP(RIGHT COL.)

CSP solver and employs the MCHC, described in Fig. 2, as Hill-Climber. If the MCHC fails the evaluation improvement of the solution 100 times succeedingly, a new initial solution is randomly generated again. The population size of SSGA and H-GA are set to be 200, which is a resultant of parameter tuning. The mutation probability of SSGA is set to be 0.025 which is identical to the 1 / (string length). We use the H-GA version 1 which shows the best performance in [3] to our experiment. The population size M of EDAs including the proposed methods is set to be 3000. The size N of selected individual used to estimate the probabilistic model is set to be 1000.

The experimental results for solving general CSPs are plotted in Fig. 5 and 6. The x and y axes of all graphs in these figures denote the values of density and tightness, respectively. All graphs in this paper are averaged results over

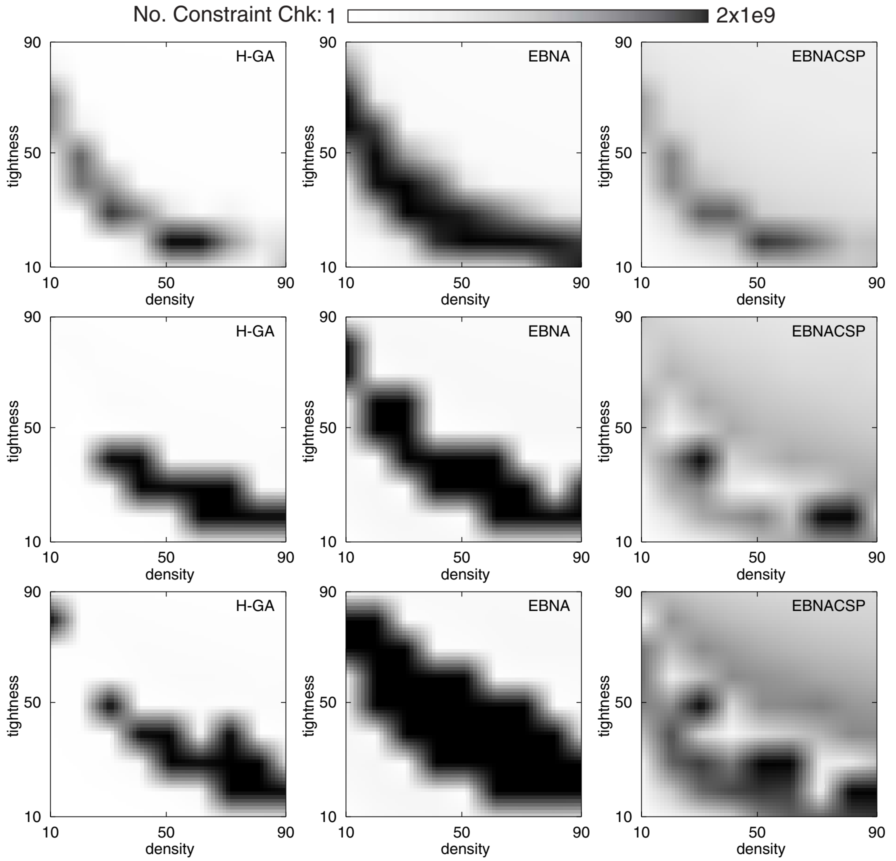

Fig. 8. The scalability of each algorithm, the number of constraint checks until finding satisfiable solutions when satisfiable solutions are found within 2 billion constraint checks: the depiction manner of each graph is the same as Fig. 6; 40 variables with 10(UPPER ROW), 20(MIDDLE ROW), and 30(LOWER ROW) labels in each variable; H-GA(LEFT COL.), EBNA(MIDDLE COL.), and EBNACSP(RIGHT COL.)

25 experiments for each couple of (density, and tightness), where density and tightness separately range from 10 to 90 with step size 10. We define “a success experiment” whether algorithms can find satisfiable solution until 2 billion constraint checks. That is, the ratio of success (Fig. 5) indicates how many success experiments exist over 25 experiments. If the success ratio is equal to 0, the number of constraint checks until finding satisfiable solution in the case of success experiments (Fig. 6) is set to be 2 billion. The colors in Fig. 5 and 6, such that favourable results tend to be brighter, and unfortunate results tend to be darker, indicate the success ratio and the number of constraint checks until finding satisfiable solutions. As shaded in all graphs, darker areas lie across the

couples of density and tightness. Such an area is called the "phase transition", where it is difficult to solve the problem for any algorithms.

As depicted in Fig. 5 and Fig. 6, the restart method and the SSGA were not found satisfiable solution effectively. Almost all experiments of them were failed. We can confirm the role of GEAs and EDAs in these hybrid approaches since naive iteration cannot help us to find satisfiable solutions. The conventional EDAs outperform conventional GA (SSGA). By comparing H-GA with SSGA, we can confirm the effectiveness of the repair algorithm for CSPs since H-GA also adopts the steady-state selection. The proposed method outperform any other algorithms in the sense of the success ratio. However they have a drawback that it takes a large number of constraint checks even if it is easier problem instances for H-GA and the proposed methods, for problem instances in the phase transition area, the proposed method outperform the H-GA.

Next, we investigate the scalability of the H-GA, EBNA and EBNACSP in Fig. 7 and Fig. 8. In these figures, the domain size (the number of labels in each variables) of general CSPs with 40 variables change from 10 to 30. EBNACSP exhibit the best scalability in the sense of the success ratio (Fig. 7) among three algorithms. H-GA can quickly solve problem instances, but its success ratio is low.

6 Conclusions

In this paper, we proposed the hybrid method of EDAs with repair method for solving CSPs. As the same as the conventional GEAs, the repair method improve the search methods of EDAs dramatically. Moreover, the proposed method outperforms the restart method, GEAs with or w/o the repair method and conventional EDAs. Future work is (1) to reduce the number of constraint checks in the case of easier problems. That is, actually, the proposed method apply the repair method to all individuals. It yields great number of constraint checks. Furthermore, (2) another one is further incorporation of the domain knowledge in CSPS to estimation of the probabilistic model. The proposed method employed BIC as the score metric. However, the BIC does not take account into the local consistency of labels. We consider the local consistency of labels makes the score metric more effective.

References

1. E. Tsang: *Foundations of Constraint Satisfaction*, Academic Press, 1993
2. J. Paredis: Genetic State-Space Search for Constraint Optimization Problems, *Proceedings of the 13th International Joint Conference on Artificial Intelligence* **II** (1993) 967–972
3. A. E. Eiben *et al.*: Solving Constraint Satisfaction Problems Using Genetic Algorithms, *Proceedings of the 1st International Conference on Evolutionary Computation* **II** (1994) 542–547

4. A. E. Eiben *et al.*: How to Apply Genetic Algorithms to Constrained Problems, *Practical Handbook of Genetic Algorithms*, Lance Chambers Editors, CRC Press, Chapter 10 (1995) 307–365
5. M.-C. Riff: Using the knowledge of the Constraint Network to design an evolutionary algorithm that solves CSP, *Proceedings of the 3rd International Conference on Evolutionary Computation* (1996) 279–284
6. M.-C. Riff: Evolutionary Search guided by the Constraint Network to solve CSP, *Proceedings of the 4th International Conference on Evolutionary Computation* (1997) 337–342
7. E. Marchiori: Combining Constraint Processing and Genetic Algorithms for Constraint Satisfaction Problems, *Proceedings of the Seventh International Conference of Genetic Algorithm* (1997) 330–337
8. G. Dozier *et al.*: Solving constraint satisfaction problems using hybrid evolutionary search, *IEEE Transactions on Evolutionary Computation* **2(1)** (1998) 23–33
9. H.Handa *et al.*: Coevolutionary Genetic Algorithm for Constraint Satisfaction with a Genetic Repair Operator for Effective Schemata Formation, *Proceedings of the 1999 IEEE System Man and Cybernetics Conference* **III** (1999) 617–621
10. S. Minton *et al.*: Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems, *Constraint-Based Reasoning*, E. C. Freuder and A. K. Mackworth Editors, The MIT Press (1994) 161–206,
11. H. Mühlenbein and G. Paaß:From Recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature - PPSN IV* (1996) 178–187
12. P. Larrañaga and J. A. Lozano Editors: *Estimation of Distribution Algorithms*, Kluwer Academic Publishers (2002)
13. Memetic Algorithms' Home Page, http://www.ing.unlp.edu.ar/cetad/mos/memetic_home.html
14. M. Pelikan: Bayesian optimization algorithm: From single level to hierarchy, Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, IL. Also IlliGAL Report No. 2002023 (2002)
15. M. Pelikan *et al.*: BOA: The Bayesian optimization algorithm, *Proceedings of the Genetic and Evolutionary Computation Conference* **1** (1999) 525–532
16. H. Mühlenbein and T. Mahnig: FDA - a scalable evolutionary algorithms for the optimization of additively decomposed functions, *Evolutionary Computation* **7(4)** (1999) 353–376
17. J. S. De Bonet *et al.*: MIMIC: Finding optima by estimating probability densities, *Advances in Neural Information Processing Systems* **9** (1996)
18. S. Baluja: Using a priori knowledge to create probabilistic models for optimization *International J. of Approximate Reasoning*, **31(3)** (2002) 193–220
19. P. Larrañaga *et al.*: Combinatorial Optimization by Learning and Simulation of Bayesian, *Uncertainty in Artificial Intelligence, Proceedings of the Sixteenth Conference* (2000) 343–352
20. E. Bengoetxea *et al.*: Learning and simulation of Bayesian networks applied to inexact graph matching, *Pattern Recognition*, **35(12)** (2002) 2867–2880
21. G. Syswerda: A Study of Reproduction in Generational and Steady-State Genetic Algorithms, *Foundations of Genetic Algorithms*, G. J. E. Rawlins Editors, Morgan Kaufman (1990) 94–101