Effects of Discrete Hill Climbing on Model Building for Estimation of Distribution Algorithms

Wei-Ming Chen
Chu-Yu Hsu
Tian-Li Yu
Wei-Che Chien

TEIL Technical Report No. 2013005
January, 2013

Taiwan Evolutionary Intelligence Laboratory (TEIL)
Department of Electrical Engineering
National Taiwan University
No.1, Sec. 4, Roosevelt Rd., Taipei, Taiwan

http://teil.ee.ntu.edu.tw

Effects of Discrete Hill Climbing on Model Building for Estimation of Distribution Algorithms

Wei-Ming Chen, Chu-Yu Hsu, Tian-Li Yu, Wei-Che Chien
National Taiwan University
r00921044@ntu.edu.tw, r01921096@ntu.edu.tw, tianliyu@cc.ee.ntu.edu.tw, r01921081@ntu.edu.tw

Abstract

Hybridization of global and local searches is a well-known technique for optimization algorithms. On estimation of distribution algorithms (EDAs), hill climbing strengthens the signals of dependencies on correlated variables and improves the quality of model building, which reduces the required population size and convergence time. However, hill climbing also consumes extra computational time. In this paper, analytical models are developed to investigate the effects of combining two different hill climbers with the extended compact genetic algorithm and the dependency structure matrix genetic algorithm. By using the one-max problem and the 5-bit non-overlapping trap problem as the test problems, the performances of different hill climbers are compared. Both analytical models and experiments reveal that the greedy hill climber reduces the number of function evaluations for EDAs to find the global optimum.

1 Introduction

Genetic and evolutionary algorithms (GEAs) have been successfully applied to a wide range of optimization problems. Some techniques (6) have been proposed to improve the efficiency of GEAs, such as parallelization, time utilization, evaluation relaxation, and hybridization. Hybridization, a combination of global and local search, is the main focus in this paper. Several studies (17; 13; 9; 27) discussed that how hybridization combines with GEAs. Some researches (22; 19; 29; 32; 2) also showed that local search enhances the performances of the estimation of distribution algorithms (EDAs) on a wide range of test problems.

One of the most commonly used local search methods is hill climbing. For EDAs, gene-wise hill climbers (16; 21) and building-block-wise hill climbers (23; 24; 15) are often investigated. The purpose of hill climbing is to push the individuals in population to local optima and to reduce the fitness variance of the population (21). Hill climbing benefits EDAs by reducing the collateral noise, which strengthens the signals of dependencies on correlated variables (4) and increases the model accuracy of EDAs.

However, performing hill climbing always consumes extra computational time. This paper focuses on investigating the overall performances for hill climbers on decomposable test problems. By using the extended compact genetic algorithm (12) (eCGA) and the dependency structure matrix genetic algorithm (30) (DSMGA) as the algorithms, the performances of EDAs in terms of number of function evaluations (NFE) with different hill climbers are analyzed.

This paper continues as follows. Section 2 describes the methodologies. Section 3 discusses the performance of EDAs without hill climbing. Sections 4 and 5 discuss the performances of steepest-descent hill climber and greedy hill climber on EDAs, respectively. Section 6 compares EDAs with hill climbers and the original EDAs. Finally, Section 7 concludes this paper.

2 Methodologies

This section describes the test problems and the hill climbers used in this paper. Section 2.1 describes the test problems. Section 2.2 describes two hill climbers, the steepest-descent hill climber (SHC) and the greedy hill climber (GHC). Sections 2.3 and 2.4 investigate the required NFE for SHC and GHC to push a binary string to local optimum on Trap-5, respectively.

2.1 Test Problems

We use the one-max problem and the 5-bit non-overlapping trap problem as the test problems in this paper. For simplicity, the one-max problem is called OneMax and the 5-bit non-overlapping trap problem is called Trap-5 in this paper. The fitness of OneMax is defined as the sum of the bits in the input binary string:

$$f_{OneMax}(X) = \sum_{i=0}^{\ell-1} X_i, \tag{1}$$

where $X = (X_0, \cdots, X_{\ell-1})$ is the input binary string of length ℓ. OneMax is a simple linear problem with the optimum being the binary string of all ones.

Trap-5 (1; 5) is the concatenated traps of order 5. Trap-5 divides a binary string into non-overlapping subproblems of 5 bits each. The fitness function of each subproblem can be written as

$$f_{trap5}(x) = \begin{cases} 5 & \text{if } x = 5 \\ 4 - x & \text{otherwise,} \end{cases} \tag{2}$$

where x is the number of ones in a subproblem. The fitness of an entire binary string is the summation of all the fitness of subproblems. Since every subproblem is a trap function, EDAs need to learn the problem structure correctly to solve Trap-5.

Algorithm 1: Steepest-descent Hill Climber

Input: The original population P
Output: The population after SHC performed, $P^{'}$

```
P' ← P
for each individual I ∈ P' do
    while a bit-flip exists to increase the fitness of I do
        Perform the bit-flip that yields the maximum fitness improvement.
return P'
```

Algorithm 2: Greedy Hill Climber

Input: The original population P
Output: The population after GHC performed $P^{'}$

```
P' ← P
for each individual I ∈ P' do
    while a bit-flip exists to increase the fitness of I do
        Randomly flips a bit that increases the fitness.
return P'
```

2.2 Steepest-descent Hill Climber and Greedy Hill Climber

Steepest-descent hill climber (SHC) and greedy hill climber (GHC) are applied to EDAs in this paper. Both SHC and GHC choose a bit to flip in each iteration. Among all bits in the binary string, SHC flips the bit that increases the fitness most. GHC randomly flips a bit that increases the fitness. Both SHC and GHC continue the iteration until no single bit-flip yields fitness improvement. Algorithms 1 and 2 show the pseudo codes of SHC and GHC, respectively.

In this paper, we compare the performances in terms of NFE for the following three cases: (1) the original EDAs, (2) EDAs with SHC, (3) EDAs with GHC. In cases that an EDA is combined with a hill climber, hill climbing is applied to the entire population in the first generation right after initialization. We perform hill climbing only once because the purpose of hill climbing is to enhance the model building. If the accurate model has been built in the first generation, crossover does not disrupt the hill-climbed subsolutions. Therefore, performing hill climbing in later generation is unnecessary since it does not flip any bit in the binary string.

2.3 NFE requirement for SHC on Trap-5

In this paper, we use NFE_{SHC} and NFE_{GHC} to denote the NFE required for SHC and GHC respectively. On Trap-5, when performing SHC on a binary string, the NFE required for each iteration is equal to the problem size. Therefore, the NFE required for SHC on a binary string equals the number of iterations times the problem size. Assume that a test problem is composed of m non-overlapped subproblems. Because the problem size and the number of flips are both proportional to m, the required NFE is proportional to m^2. Therefore, if the population size is n, the required NFE for SHC can be expressed as

$$NFE_{SHC} = \Theta(m^2 n). \tag{3}$$

2.4 NFE requirement for GHC on Trap-5

Assume that a binary string with the problem size ℓ has b bits that increases the fitness value after flip one of the bits. Let the expectation of the number of trials for finding a bit to improve the fitness value after the bit is flipped to be $E_{try}(\ell, b)$. The probability that flipping the first chosen bit increases the fitness value is b/ℓ. The following recursive function holds.

$$\begin{aligned} E_{try}(\ell, b) &= 1 \times \frac{b}{\ell} + (1 + E_{try}(\ell - 1, b)) \times \frac{\ell - b}{\ell} \\ &= 1 + \frac{\ell - b}{\ell} \times E_{try}(\ell - 1, b). \end{aligned} \tag{4}$$

With the boundary condition that $E_{try}(b, b)$ is equals to 1, Equation 4 can be derived as

$$E_{try}(\ell, b) = \frac{\ell + 1}{b + 1}. \tag{5}$$

Flipping b bits needs b iterations. The expectation of required NFE for one individual can be calculated as

$$\begin{aligned} NFE_{GHC} &= E_{try}(\ell, b) + E_{try}(\ell, b - 1) + \cdots + E_{try}(\ell, 1) \\ &= \frac{\ell + 1}{b + 1} + \frac{\ell + 1}{b} + \cdots + \frac{\ell + 1}{2} \\ &= \Theta(\ell \log b). \end{aligned} \tag{6}$$

Since ℓ and b are both proportional to m, given the population size n, the required NFE for GHC can be also expressed as

$$NFE_{GHC} = \Theta(mn \log m). \tag{7}$$

3 EDA without Hill Climbing

The NFE for performing an EDA without hill climbing is generally equal to n times the number of generations. In EDAs, previous work showed that the required population for building the model accurately is proportional to $m \log m$ on nearly decomposable problems (31). Because the restricted tournament replacement (11) is performed and the terminal condition is the appearance of global optimum instead of the convergence of population, Thieren's convergence time model (28) can not be applied to our work. Since the theorems can not be applied to calculate the NFE, we use the result from experiments instead.

Experiments of the required NFE on DSMGA are shown in Figures 1 and 2. On both eCGA and DSMGA, the results show that the NFE is proportional to $m^{1.3}$ on OneMax and is proportional to m^2 on Trap-5. This paper uses the bisection method (22) to estimate an optimal population size for every test problem. A population size is considered enough when the EDA finds out the global optimum for 10 consecutive successful runs. The required population size is determined as the average over 50 bisection runs. The average NFE over 500 runs is recorded under the required population size.

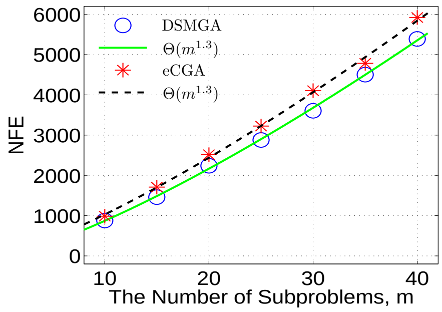

Figure 1: The relations between m and NFE with eCGA and DSMGA on OneMax. NFE is proportional to $m^{1.3}$. Hill climbing is not performed.

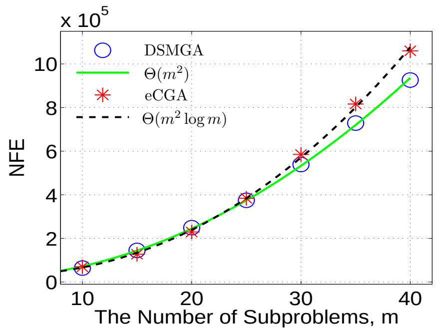

Figure 2: The relations between m and NFE with eCGA and DSMGA on Trap-5. NFE is proportional to $m^2 \log m$ on eCGA and proportional to m^2 on DSMGA. Hill climbing is not performed.

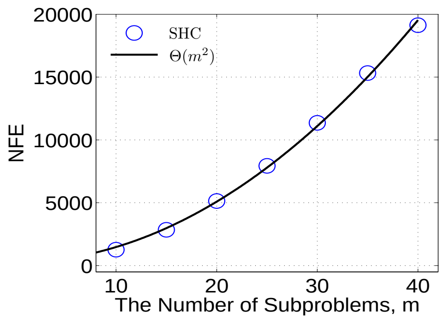

Figure 3: The relation between m and NFE_{SHC} with SHC on OneMax. NFE_{SHC} is proportional to m^2.

4 EDA with SHC

SHC pushes the binary string to the global optimum on OneMax regardless of the problem length. Specifically, a population with only one individual is enough to solve the problem. Therefore, according to Equation 3, the required NFE is proportional to m^2. Figure 3 shows the empirical results.

On Trap-5, after performing SHC, every subproblem in a binary string only contains 00000 or 11111. Let g_1 and g_2 be two dependent genes. The joint probability distribution of g_1 and g_2 is

$$P_{g_1 g_2} = \begin{cases} 6/32 & g_1 g_2 = 11 \\ 0 & g_1 g_2 = 10 \\ 0 & g_1 g_2 = 01 \\ 26/32 & g_1 g_2 = 00. \end{cases} \tag{8}$$

The reason is that SHC forces subproblems which contains no less than four 1s into five 1s, and forces the others into zero 1s.

Binary tournament selection is applied to the population after performing hill climbing. Because only two local optima (00000 or 11111) exist for a subproblem after SHC, Trap-5 with m subproblems can be transformed to OneMax with problem size m. Assume that p_t is the probability that a subproblem containing five 1s before selection and p_{t+1} is the probability that a subproblem contains five 1s after selection. According to Thieren's performance model (28) with perfect mixing, the change in the probability is

$$p_{t+1} - p_t = \frac{I}{\sqrt{m}} \sqrt{p_t(1-p_t)}. \tag{9}$$

The value of selection intensity I is equal to $\frac{1}{\sqrt{\pi}}$ (3) if binary tournament selection is applied.

The probability of a subproblem that contains five 1s after selection can be written as

$$\begin{aligned} p_{t+1} &= \frac{3}{16} + \frac{1}{\sqrt{m\pi}} \sqrt{\frac{3}{16}\left(1 - \frac{3}{16}\right)} \\ &\approx 0.1875 + \frac{0.22}{\sqrt{m}}. \end{aligned} \tag{10}$$

To sum up, the joint probability distribution of g_1 and g_2 after SHC and selection is

$$P_{g_1 g_2} \approx \begin{cases} 0.1875 + 0.22/\sqrt{m} & g_1 g_2 = 11 \\ 0 & g_1 g_2 = 10 \\ 0 & g_1 g_2 = 01 \\ 0.8125 - 0.22/\sqrt{m} & g_1 g_2 = 00. \end{cases} \tag{11}$$

To verify the above derivations, several experiments are conducted by fixing n to 1000. For each m, the results are averaged over 10000 independent experiments. A pair of dependent genes is chosen randomly on every individual in each experiment and the value of these genes after hill climbing and selection is recorded. Compared with the joint probability distribution shown in Figure 4, the empirical results verify the derivations.

For model building, one of the commonly used metrics is Shannon's entropy (25). The entropy is defined as $H(X) = \sum_{i=1}^{n} p(x_i) \lg(p(x_i))$, where $p(x_i)$ is the probability mass function of outcome x_i.[1] Both eCGA and DSMGA use Shannon's entropy to detect the dependency between variables. After performing hill climbing, the joint probability distribution for calculating Shannon's entropy on dependent genes is shown as Equation 11. M_0 and M_1 are defined as the mutual information between independent genes and dependent genes after hill climbing and selection. Let $\hat{M}_{0,n}$ and $\hat{M}_{1,n}$ denote the sampled mutual information for M_0 and M_1 respectively, where n is the population size.

[1] We use the symbol lg for the binary logarithm in this paper.

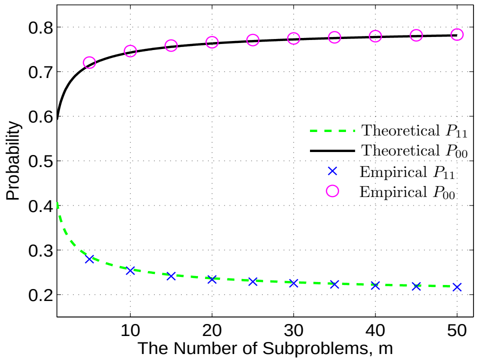

Figure 4: The theoretical data and the empirical data of the joint probability distributions on different m. Empirical results verify the theoretical equations.

Hutter and Zaffalon (14) defined J and K for calculating the mean and the variance of the sampled mutual information. The equations can be written as

$$J = \sum_{ij} p_{ij} \lg \frac{p_{ij}}{p_i p_j}. \tag{12}$$

$$K = \sum_{ij} p_{ij} \left(\lg \frac{p_{ij}}{p_i p_j} \right)^2. \tag{13}$$

After substituting Equation 11 into Equation 12, the following equation holds.

$$\begin{aligned} J = \quad & -\left(0.1875 + \tfrac{0.22}{\sqrt{m}}\right) \lg \left(0.1875 + \tfrac{0.22}{\sqrt{m}}\right) \\ & -\left(0.8125 - \tfrac{0.22}{\sqrt{m}}\right) \lg \left(0.8125 - \tfrac{0.22}{\sqrt{m}}\right). \end{aligned} \tag{14}$$

When variable a is much greater than variable b, $\lg(a+b)$ can be approximated to $\lg a + b/a$, assuming m is large enough, Equation 14 can be simplified as

$$\begin{aligned} J = & -\left(0.1875 + \frac{0.22}{\sqrt{m}}\right)\left(\lg(0.1875) + \frac{0.22}{0.1875\sqrt{m}}\right) \\ & -\left(0.8125 - \frac{0.22}{\sqrt{m}}\right)\left(\lg(0.8125) - \frac{0.22}{0.8125\sqrt{m}}\right) \\ \approx & \frac{c_1}{\sqrt{m}} + c_2, \end{aligned} \tag{15}$$

where the term $\mathcal{O}(1/m)$ is neglected. Positive constants are denoted as c_i for simplicity in this section. Similarly, Equation 13 can be simplified as

$$\begin{aligned} K = & -\left(0.1875 + \frac{0.22}{\sqrt{m}}\right)\left(\lg(0.1875) + \frac{0.22}{0.1875\sqrt{m}}\right)^2 \\ & -\left(0.8125 - \frac{0.22}{\sqrt{m}}\right)\left(\lg(0.8125) - \frac{0.22}{0.8125\sqrt{m}}\right)^2 \\ \approx & -\frac{c_3}{\sqrt{m}} + c_4. \end{aligned} \tag{16}$$

Based on Equations 15 and 16, the means and the variances of the sampled mutual information

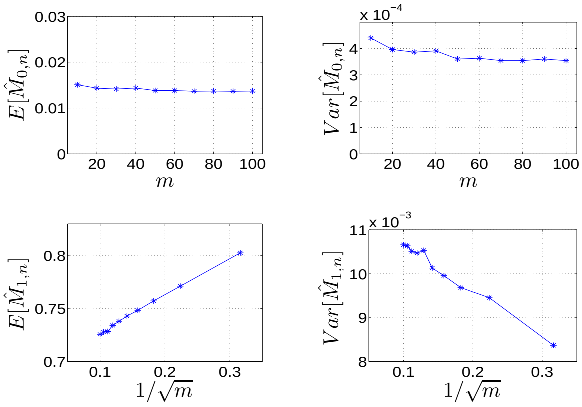

Figure 5: The effect of different problem sizes on the sampled mutual information. The population size is fixed to 100. For the pair of the independent genes, the sampled mutual information and the variance of sampled mutual information are both virtually independent of m. That for the pair of dependent genes, there is a linear relation.

for a finite number of samples can be derived as

$$E[\hat{M}_{0,n,1,m}] = \frac{1}{2n} + \mathcal{O}(\frac{1}{n^2}). \tag{17}$$

$$Var[\hat{M}_{0,n,1,m}] = \frac{1}{2n^2} + \mathcal{O}(\frac{1}{n^3}). \tag{18}$$

$$E[\hat{M}_{1,n,1,m}] = \frac{c_1}{\sqrt{m}} + c_2 + \frac{1}{2n} + \mathcal{O}(\frac{1}{n^2}). \tag{19}$$

$$Var[\hat{M}_{1,n,1,m}] = \frac{-\frac{c_3}{\sqrt{m}} + c_4}{n} + \mathcal{O}(\frac{1}{n^2}). \tag{20}$$

Equations 17 and 18 indicate that both $E[\hat{M}_{0,n}]$ and $Var[\hat{M}_{0,n}]$ are independent of m. Equation 19 shows that there is a positive linear relation between $1/\sqrt{m}$ and $E[\hat{M}_{1,n}]$. Equation 20 indicates a negative linear relation between $1/\sqrt{m}$ and $Var[\hat{M}_{1,n}]$. To verify the above equations, experiments are conducted with different m. The population size is fixed to 100. All results are averaged over 10000 independent runs. The empirical results in Figure 5 agree with the above indications.

After modeling the means and the variances of the sampled mutual information for a finite population, the population-sizing model can be derived by the decision-making approach. By approximating the distribution of the sampled mutual information as a Gaussian distribution (14), the decision making error can be calculated as follows. Define a variable τ as

$$\tau \triangleq \frac{E[Z]}{\sqrt{Var[Z]}} \tag{21}$$

where Z $= \hat{M}_{1,n} - \hat{M}_{0,n}$. The decision error ϵ is given by $1 - \Phi(\tau)$. Given Equations 17 to 20, Equation 21 can be transformed to

$$\tau \triangleq \frac{\frac{c_1}{\sqrt{m}} + c_2}{-\frac{c_3}{\sqrt{m}} + c_4}\sqrt{n}. \tag{22}$$

When m is large enough, $c_1/\sqrt{m}$ and $-c_3/\sqrt{m}$ can be neglected, and hence τ can be viewed as

$$\tau \approx c_5\sqrt{n}. \tag{23}$$

For a large τ, ϵ can be approximated as

$$\epsilon \approx \frac{1}{\tau}e^{-\frac{\tau^2}{2}} \approx \frac{c_6}{\sqrt{n}}e^{-c_7 \cdot n}. \tag{24}$$

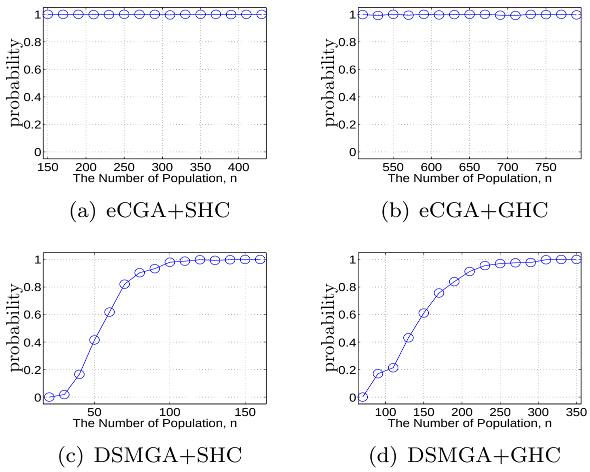

Figure 6: The probability of different EDAs to solve Trap-5 when the accurate model is learned. The number of subproblems is fixed to 20. When the accurate model is learned, eCGA solves Trap-5 but DSMGA fails sometimes.

For a problem with m subproblems, one subproblem can be treated as one decision variable, and $\binom{m}{2}$ independent decisions need to be made correctly (31).[2] Given the model accuracy to be $(1 - 1/m)$, the following relation holds.

$$(1-\epsilon)^{\Theta(m^2)} \geq 1 - \frac{1}{m}. \tag{25}$$

For a small ϵ and a large m, Inequality 25 can be simplified as

$$\frac{1}{\epsilon} \geq \Theta(m^3). \tag{26}$$

With arithmetic manipulations, the following relation holds.

$$\lg(\frac{\sqrt{n}}{c_6}) + c_7 \cdot n \geq \Theta(\lg m). \tag{27}$$

For a large n, the first term in Equation 27 can be neglected. The following bound is obtained.

$$n \geq c_8 \cdot \lg m. \tag{28}$$

The population-sizing requirements for EDAs fall into three major categories (18): (1) building block supply (8), (2) decision making (7; 10), and (3) model building (31). When an EDA is concerned alone, the population-sizing requirements for model building is the upper bound of the other two (31). SHC reduces the population-sizing requirements for model building. For DSMGA with SHC, even the population size is large enough to learn the accurate model, the global optimum of Trap-5 is not always found (Figure 6(c)). eCGA with SHC always solves Trap-5 when the population size is large enough to learn the accurate model (Figure 6(a)).

Figure 7 shows the empirical results on EDAs with SHC on Trap-5. n is proportional to $\log m$ on eCGA since the required population is still bounded by the accuracy of model building. On DSMGA, n is proportional to $\sqrt{m}$. As the result, NFE_{SHC} can be calculated according to Equation 3. NFE_{SHC} is proportional to $m^2 \log m$ on eCGA and proportional to $m^{2.5}$ on DSMGA. Figure 8 shows the relations between NFE and m, which verify the derivations.

5 EDA with GHC

GHC pushes the binary string to the global optimum on OneMax regardless of the problem length, and a population with only one individual is enough to solve the problem. According to Equation 7, the required NFE is proportional to $m \log m$. Figure 9 shows the empirical results.

[2] $\binom{a}{b}$ is the coefficient of the x^b term in the polynomial expansion of the binomial power $(1 + x)^a$.

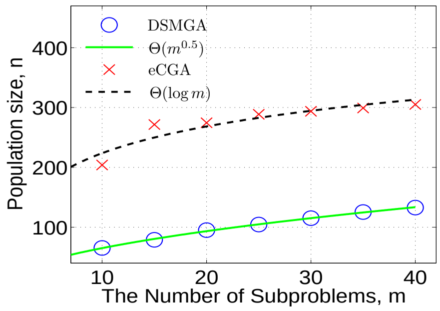

Figure 7: The relations between m and n on EDAs with SHC on Trap-5. n is proportional to $\sqrt{m}$ on DSMGA. Derivations show that n is proportional to $\log m$ on eCGA but empirical results do not match well.

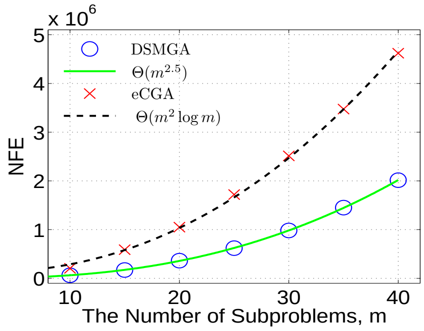

Figure 8: The relations between m and NFE_{SHC} on EDAs with SHC on Trap-5. NFE_{SHC} is proportional to $m^2 \log m$ on ecGA and proportional to $m^{2.5}$ on DSMGA.

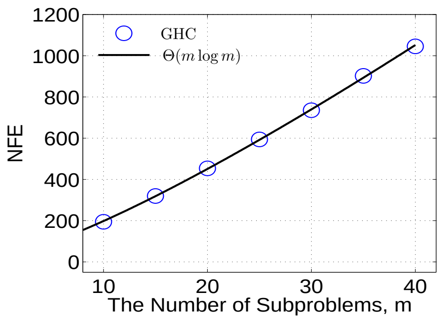

Figure 9: The relation between m and NFE_{GHC} with GHC on OneMax. NFE_{GHC} is proportional to $m \log m$.

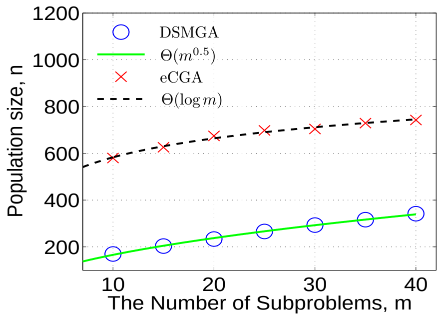

Figure 10: The relations between m and n on EDAs with GHC on Trap-5. n is proportional to $\log m$ on eCGA and proportional to $\sqrt{m}$ on DSMGA.

On Trap-5, GHC pushes the subproblems with no more than three 1s to 00000. For the subproblems with four 1s, the probability that GHC pushes the subproblems to 11111 is 1/5. Therefore, the joint probability distribution of g_1 and g_2 is

$$P_{g_1g_2} = \begin{cases} 2/32 & g_1g_2 = 11 \\ 0 & g_1g_2 = 10 \\ 0 & g_1g_2 = 01 \\ 30/32 & g_1g_2 = 00. \end{cases} \tag{29}$$

Recall Equation 9 in Section 4, the probability of a subproblem that contains five 1s after selection can be written as

$$\begin{aligned} p_{t+1} &= \frac{1}{16} + \frac{1}{\sqrt{m\pi}}\sqrt{\frac{1}{16}\left(1 - \frac{1}{16}\right)} \\ &\approx 0.0625 + \frac{0.14}{\sqrt{m}}. \end{aligned} \tag{30}$$

The joint probability distribution of g_1 and g_2 after selection is

$$P_{g_1g_2} \approx \begin{cases} 0.0625 + 0.14/\sqrt{m} & g_1g_2 = 11 \\ 0 & g_1g_2 = 10 \\ 0 & g_1g_2 = 01 \\ 0.9375 - 0.14/\sqrt{m} & g_1g_2 = 00. \end{cases} \tag{31}$$

After substituting Equation 31 into Equations 12 and 13, we can also derive similar equations to Equations 17 to 20. Thus, the needed population size for model building is proportional to $\log m$ when m is large enough. When EDAs accurately learn the model, we examine the probability of finding out the global optimum on Trap-5. DSMGA fails sometimes (Figure 6(d)), and eCGA succeeds in almost every cases (Figure 6(b)). Therefore, n is proportional to $\log m$ on eCGA since the required population is still bounded by the accuracy of model building. Empirical results in Figure 10 agree the derivations.

NFE_{GHC} can be calculated by combining the results above with Equation 7. NFE_{GHC} is proportional to $m \log^2 m$ on eCGA and proportional to $m^{1.5} \log m$ on DSMGA. Figure 11 shows the relations between NFE and m, and the empirical results verify the derivations.

6 Discussions

On OneMax, SHC and GHC always push every binary string in the population to the global optimum, and hence one binary string is enough to solve the problem. Comparing with GHC, SHC needs to examine every bit-flip before deciding which bit to flip. Therefore, GHC outperforms SHC in terms of NFE. Empirical results show that GHC consumes fewer NFE than eCGA, DSMGA and SHC (Figure 12).

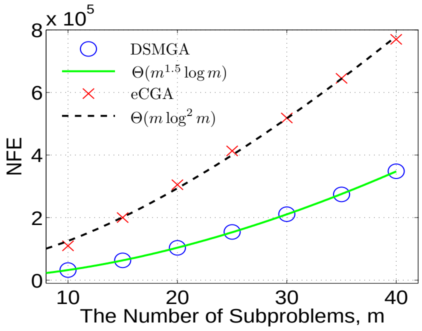

Figure 11: The relations between m and NFE_{GHC} on EDAs with GHC on Trap-5. NFE_{GHC} is proportional to $m \log^2 m$ on eCGA and proportional to $m^{1.5} \log m$ on DSMGA.

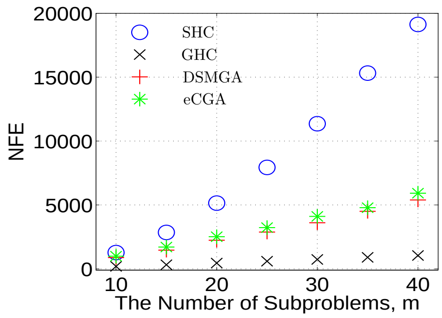

Figure 12: SHC and GHC can solve OneMax without EDAs. The comparison of the required NFE of hill climbers and EDAs on OneMax is shown. GHC consumes fewer NFE than the others.

On Trap-5, among the three population-sizing requirements (building block supply, decision making, and model building) (18), model building bounds the other two when EDA is concerned alone (31). However, when combining with hill climbing, the required population for model building is reduced. To determine whether the required population for model building is still the upper bound of the other two, we record the probability of finding the global optimum when the accuracy model is learned.

For eCGA, empirical results show that when the accuracy model is learned, the global optimum can be found (Figures 6(a) and 6(b)). That is, the required population for model building is still the upper bound of the other two. Therefore, according to Equation 28, n is proportional to $\log m$. Combining Equation 28 with Equations 3 and 7, NFE is proportional to $(m^2 \log m)$ and $(m \log^2 m)$ on eCGA with SHC and GHC, respectively. Figure 13 shows the comparison of the NFE on eCGA with different hill climbers.

For DSMGA, finding the accurate model does not lead to solve the problem (Figures 6(c) and 6(d)). Empirical results show that n is proportional to $\sqrt{m}$ (Figures 7 and 10). The population-sizing models on DSMGA with SHC and on DSMGA with GHC have the similar growth rates on different m, GHC outperforms SHC because GHC consumes fewer NFE. The comparisons between DSMGA without a hill climber, DSMGA with SHC, DSMGA with GHC is shown in Figure 14.

7 Conclusion

This paper investigated the effect of hill climbing for EDAs on OneMax and Trap-5. Two different hill climbers were considered in this work. Our derivations showed that hill climbing improved the accuracy of model building, which reduced the required population size to solve the problems. To verify our derivations, the performances of EDAs were compared with and without hill climbing in terms of NFE. Empirical results showed that performing EDAs with GHC reduced the NFE on both OneMax and Trap-5, while SHC consumed more NFE when combined with EDAs.

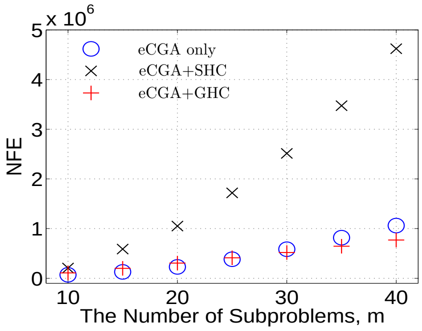

Figure 13: On Trap-5, eCGA with GHC outperforms eCGA with SHC and eCGA without hill climbers.

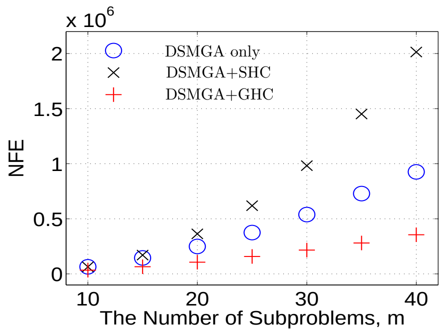

Figure 14: On Trap-5, DSMGA with GHC outperforms DSMGA with SHC and DSMGA without hill climbers.

Hill climbers strengthen the signals of dependencies on correlated variables by pushing subproblems to either global or local optimum. As the result, the required population to solve the problem is reduced. However, hill climbers consume additional NFE, which should be taken into consideration. This paper focuses on two commonly used hill climbers, SHC and GHC. The order of NFE on EDAs with greedy hill climber ($\Theta(mn \log m)$) is less than that with steepest-descent hill climber ($\Theta(m^2 n)$). Therefore, EDAs with greedy hill climber consume fewer NFE to solve the proposed test problems.

Several directions of future work are described as follows. This paper shows that greedy hill climber benefits EDAs on both OneMax and Trap-5. Compared with the decomposable test problems, flipping a bit may benefit two or more subproblems on the overlapping test problems like the Ising spin glasses (26) or the NK-landscapes with nearest neighbor (20). Therefore, hill climbers may benefit EDAs more on the problems with overlapping substructures. Such analysis would be a topic for future work. In this paper, we considered only two hill climbers. The analysis of other local search methods on EDAs is another possible future work.

This paper investigates the effects of two different hill climbers on EDAs. In addition to these two hill climbers, other local searchers may be also considered, and we believe that they can be analyzed in a similar manner in this paper. Such analysis is essential to design specialized local searcher for EDAs in the future.

References

[1] D. H. Ackley. An empirical study of bit vector function optimization. *Genetic Algorithms ans Simulated Annealing*, pages 170–204, 1987.

[2] U. Aickelin, E. K. Burke, and J. Li. An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society*, 58(12):1574–1585, 2007.

[3] T. Blickle and L. Thiele. A mathematical analysis of tournament selection. *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA'95)*, pages 9–16, 1995.

[4] P. A. Bosman and D. Thierens. The roles of local search, model building and optimal mixing in evolutionary algorithms from a bbo perspective. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2011)*, pages 663–670, 2011.

[5] K. Deb and D. E. Goldberg. Sufficient conditions for deceptive and easy binary functions. *Annals of Mathematics and Artificial Intelligence*, 10(4):385–408, 1993.

[6] D. E. Goldberg. *The design of innovation: Lessons from and for competent genetic algorithms.* Kluwer Academic Publishers, Boston, MA, 2002.

[7] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992.

[8] D. E. Goldberg, K. Sastry, and T. Latoza. On the supply of building blocks. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 336–342, 2001.

[9] D. E. Goldberg and S. Voessner. Optimizing global-local search hybrids. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, pages 220–228, 1999.

[10] G. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pages 7–12, 1997.

[11] G. R. Harik. Finding multimodal solutions using restricted tournament selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31, 1995.

[12] G. R. Harik, F. G. Lobo, and K. Sastry. Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA). In *Scalable Optimization via Probabilistic Modeling*, chapter 3, pages 39–61. Springer Berlin Heidelberg, Berlin, 2006.

[13] W. E. Hart. *Adaptive global optimization with local search.* PhD thesis, La Jolla, CA, USA, 1994. UMI Order No. GAX94-32928.

[14] M. Hutter and M. Zaffalon. Distribution of mutual information from complete and incomplete data. *Computational Statistics and Data Analysis*, 48(3):633–657, 2005.

[15] D. Iclănzan and D. Dumitrescu. Overcoming hierarchical difficulty by hill-climbing the building block structure. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, pages 1380–1387, 2007.

[16] C. F. Lima, M. Pelikan, F. G. Lobo, and D. E. Goldberg. Loopy substructural local search for the bayesian optimization algorithm. *Proceedings of the Second International Workshop on Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics (SLS '09)*, pages 61–75, 2009.

[17] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report C3P Report 826, California Institute of Technology, Pasadena, CA, 1989.

[18] M. Pelikan, E. Cantú-Paz, and D. E. Goldberg. Bayesian optimization algorithm, population sizing, and time to convergence. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 275–282, 2000.

[19] M. Pelikan and D. E. Goldberg. Hierarchical BOA solves Ising spin glasses and MAXSAT. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, pages 1271–1282, 2003.

[20] M. Pelikan, K. Sastry, D. E. Goldberg, M. V. Butz, and M. Hauschild. Performance of evolutionary algorithms on nk landscapes with nearest neighbor interactions and tunable overlap. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2009)*, pages 851–858, 2009.

[21] E. Radetic, M. Pelikan, and D. E. Goldberg. Effects of a deterministic hill climber on hboa. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2009)*, pages 437–444, 2009.

[22] K. Sastry. Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.

[23] K. Sastry and D. E. Goldberg. Let's get ready to rumble: Crossover versus mutation head to head. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, pages 126–137, 2004.

[24] K. Sastry and D. E. Goldberg. Let's get ready to rumble redux: crossover versus mutation head to head on exponentially scaled problems. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, pages 1380–1387, 2007.

[25] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[26] D. Sherrington and S. Kirkpatrick. Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35:1792–1796, 1975.

[27] A. Sinha and D. E. Goldberg. Verification and extension of the theory of global-local hybrids. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 591–597, 2001.

[28] D. Thierens and D. E. Goldberg. Convergence models of genetic algorithm selection schemes. *Parallel Problem Solving fron Nature (PPSN III)*, pages 119–129, 1994.

[29] S. Tsutsui, M. Pelikan, and A. Ghosh. Edge histogram based sampling with local search for solving permutation problems. *International Journal of Hybrid Intelligent Systems*, 3(1):11–22, 2006.

[30] T.-L. Yu, D. E. Goldberg, A. Yassine, and Y.-p. Chen. Genetic algorithm design inspired by organizational theory: Pilot study of a dependency structure matrix driven genetic algorithm. *Proceedings of Artificial Neural Networks in Engineering (ANNIE 2003)*, pages 327–332, 2003.

[31] T.-L. Yu, K. Sastry, D. E. Goldberg, and M. Pelikan. Population sizing for entropy-based model building in discrete estimation of distribution algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, pages 601–608, 2007.

[32] Q. Zhang, J. Sun, E. Tsang, and J. Ford. Estimation of distribution algorithm with 2-opt local search for the quadratic assignment problem. *Towards a New Evolutionary Computation*, 192:281–292, 2006.