International Journal of Innovative Computing, Information and Control
Volume **8**, Number **12**, December **2012**
 ISSN 1349-4198
pp. 8355–8366

BACKBONE GUIDED EXTREMAL OPTIMIZATION FOR THE HARD MAXIMUM SATISFIABILITY PROBLEM

Guoqiang Zeng[1], Yongzai Lu[2], Yuxing Dai[1], Zhengguang Wu[2]
Weijie Mao[2], Zhengjiang Zhang[1] and Chongwei Zheng[1]

[1]College of Physics and Electronic Information Engineering
Wenzhou University
Chashan University Town, Wenzhou 325035, P. R. China
{ zenggq; yxdai; zjzhang; zcw }@wzu.edu.cn

[2]State Key Laboratory of Industrial Control Technology
Zhejiang University
No. 38, Zheda Road, Hangzhou 310027, P. R. China
yongzailu@yahoo.com; nashwzhg@zju.edu.cn; wjmao@csc.zju.edu.cn

Received September 2011; revised January 2012

Abstract. *The original Extremal Optimization (EO) algorithm and its modified versions have been successfully applied to a variety of NP-hard optimization problems. However, there exists a problem that almost all existing EO-based algorithms have overlooked the inherent structural properties behind the optimization problems, e.g., the backbone information. This paper proposes a novel stochastic search method called "Backbone Guided Extremal Optimization (BGEO)" to solve the hard maximum satisfiability (MAX-SAT) problem, one of typical NP-hard combinatorial optimization problems. The basic idea behind BGEO is to incorporate the backbone information into EO to guide the entire search process approaching the optimal solutions. The experimental results on many reported hard MAX-SAT instances have shown the superiority of BGEO to the reported EO-based algorithm without backbone information.*
Keywords: Backbone, Extremal optimization, Maximum satisfiability problem, Phase transition

1. **Introduction.** Extremal optimization (EO) [1,2] was originally inspired by far-from-equilibrium dynamics of Bak-Sneppen model [3] of biological co-evolution, showing the features of self-organized criticality (SOC) [4]. This method provides a novel insight into the optimization domain as its novel evolutionary mechanism that merely selects against the bad, instead of favoring the good, randomly or according to a power-law distribution. Beneficial from this evolutionary mechanism, the search dynamics of the EO-based algorithms has non-equilibrium feature, which is different from other evolutionary algorithms [5]. During the past decade, the basic EO algorithm and its modified versions have been successfully applied to a variety of NP-hard optimization problems, such as graph partitioning [6], graph coloring [7], spin glasses [8,9], Lennard-Jones clusters [10], travelling salesman problem (TSP) [11-13], maximum satisfiability (MAX-SAT) problem [14] and some practical engineering problems [15-17]. The more comprehensive introductions concerning EO are referred to the surveys [18,19]. However, there exists a problem that almost all existing EO-based algorithms have overlooked the inherent structural properties behind the optimization problems, e.g., backbone information [20].

In fact, the computational complexity of an optimization problem depends on not only its dimension, but also some inherent structural properties, e.g., backbone. As one of the most interesting and important structures, backbone has been used to explain the difficulty of problem instances [20-25]. The problems with larger backbone are generally harder for local search algorithms to solve because the clustered solutions in these problems often result in these algorithms making mistakes more easily and wasting time for searching empty subspaces before correcting the bad assignments [20]. On the other hand, the utilization of the backbone information may help the design of effective and efficient optimization algorithms. For example, Schneider et al. [26,27] have developed a powerful parallel algorithm for travelling salesman problem (TSP) by using its backbone information. Dubolis and Dequen [28] incorporated the backbone information in a DPL-type algorithm for random 3-SAT problem. Telelis and Stamatopoulos [29] developed a heuristic backbone sampling method to generate initial solutions for a local search algorithm based on the concept of backbone. Zhang [30] proposed a backbone-guided WALKSAT method where the backbone information is embedded in a popular local search algorithm, such as WALKSAT. Furthermore, the basic idea has been extended to TSP [31], partial MAX-SAT problem [32]. The experimental results have shown that these backbone-based methods provide better performance than the pure local search ones.

In this paper, we focus on a novel method called 'backbone guided extremal optimization (BGEO)' for the hard MAX-SAT problem, a well-known NP-hard optimization problem. Recently, a modified EO algorithm called Bose-Einstein-EO (BE-EO) [14] has been developed to solve MAX-SAT problem. The basic idea behind BE-EO is to sample initial configurations set based on Bose-Einstein distribution to the original τ-EO search process. The experimental results on both random and structured MAX-SAT instances demonstrate BE-EO's superiority to more elaborate stochastic optimization methods such as SA [33], GSAT [34], WALKSAT [35] and Tabu search [36]. In our recent research [37], a more generalized EO framework termed as EOSAT is proposed to solve MAX-SAT problem. The modified algorithms, such as BE-EEO and BE-HEO, provide better performance than BE-EO. Therefore, by incorporating the backbone information into the EOSAT framework, the BGEO method proposed in this study is possible to guide the search approaching the optimal solutions, and to further improve the performance of the original EO algorithms.

The remainder of this paper is organized as follows. Section 2 introduces the MAX-SAT problem and some probability distributions. Then, we present the framework of BGEO in Section 3. Section 4 gives the experimental results on hard MAX-SAT instances. Finally, the conclusions and the future work are given in Section 5.

2. MAX-SAT Problem and Probability Distributions.

As the optimization counterpart of the Boolean satisfiability (SAT) problem, the MAX-SAT problem is one of well-studied NP-hard optimization problems [38].

Definition 2.1. *The weighted MAX-SAT problem can be defined to find an assignment $S = (x_1, x_2, \ldots, x_n)$ to maximize the total weight W_c ($\mathbf{C}$, $\mathbf{W}$, S) of the satisfied clauses, i.e.,*

$$\max W_c(\boldsymbol{C}, \boldsymbol{W}, S) = \max \sum_{C_i(S)=1} w_i \tag{1}$$

where $\{x_1, x_2, \ldots, x_n\}$ is a set of Boolean variables, $\boldsymbol{C} = \{C_1, C_2, \ldots, C_m\}$ is a set of clauses, each of which is a disjunction of literals l_{ij} and l_{ij} is a Boolean variable x_i or its negation, and $\boldsymbol{W} = (w_i)_{\in} N^m$ is an integer vector and w_i is the weight of the clause C_i. Clearly, the dual problem is to find an assignment S to find an assignment S to minimize the total weight $W_{cu}(\boldsymbol{C}, \boldsymbol{W}, S)$ of unsatisfied clauses, i.e.,

$$\min W_{cu}(\boldsymbol{C}, \boldsymbol{W}, S) = \min \sum_{C_j(S)=0} w_j \tag{2}$$

Remark 2.1. *If each clause consists of K literals, where K is a positive constant, we call this problem as MAX-K-SAT. Specially, when $w_i = 1$ for each clause, the problem is called unweighted MAX-SAT. It is obvious that MAX-K-SAT is more general than K-SAT, because its solution can be used to answer the question of K-SAT problem, but not vice versa [30].*

Remark 2.2. *The parameter controlling the satisfiability of an instance is $\alpha = m/n$, the ratio of clauses to variables. Extensive empirical and analytical research [21,39] have shown that for K-SAT ($K \geq 3$), the solving-cost of many local search algorithms can be characterized as "easy-hard-easy" phase transition. There exists a similar feature for MAX-K-SAT but with "easy-hard" phase transition [30].*

Originating from quantum physics, Bose-Einstein (BE) distribution [40] describes the statistical behavior of bosons (integer spin particles).

Definition 2.2. *In the context of combinatorial description [41], BE distribution can be defined as:*

$$p_x = \frac{1}{(n+1)\begin{pmatrix} n \\ x[V] \end{pmatrix}}, \quad \forall x \in \{0,1\}^n \tag{3}$$

where p_x is the probability distribution of x in the space $\{0,1\}^n$, $V = \{1, 2, \ldots, n\}$ is a base set for a given n, $x = \{x_1, x_2, \ldots, x_n\}$ is a set of Boolean variables, and $x[V] = \sum_{i \in V} x_i$ is the number of the variables of x equal to 1 in V. Its conditional probability is

$$p(x_j = 1) = \frac{x[S] + 1}{(j-1) + 2}, \text{ where } S = \{1, 2, \ldots, (j-1)\} \tag{4}$$

Definition 2.3. *The power-law $P_p(k)$, exponential distribution $P_e(k)$ and hybrid distribution $P_h(k)$ used for evolution in EO [13] are described as follows:*

$$P_p(k) = k^{-\tau}, \quad (1 \leq k \leq n) \tag{5}$$

$$P_e(k) = e^{-\mu k}, \quad (1 \leq k \leq n) \tag{6}$$

$$P_h(k) = e^{-hk} k^{-h}, \quad (1 \leq k \leq n) \tag{7}$$

where τ, μ and h all are positive constants for a specific problem with size n.

3. **Backbone Guided Extremal Optimization.** It has shown that for the MAX-SAT problem, only the BE distribution can guarantee that an initial assignment set is generated with an arbitrary proportion of 1s and 0s [41]. Moreover, the experimental results [14] on random and structured MAX-SAT instances have shown that BE-EO algorithm starting from BE-based initial configurations outperforms τ-EO from uniformly random ones. Therefore, a BE-based assignment called "BE-based Initial Configuration Generator (BEICG) [37]" will be used as the initial configuration of the proposed framework in this paper.

According to the seminal work [1,2], one of the most important issues for designing the EO-based algorithms is the appropriate definition of local and global fitness. More specifically, the global fitness (i.e., objective function) of an optimization problem should be decomposed into the local fitness (i.e., the contribution from the decision variables).

Definition 3.1. *For a given configuration S of a weighted MAX-SAT problem, the local fitness λ_i of each variable x_i is defined as follows:*

$$\lambda_i = \frac{-\sum_{x_i \in C_j \text{ and } C_j(S)=0} w_j}{\sum_{x_i \in C_k} w_k} \tag{8}$$

In other words, the local fitness is defined as the fraction of the sum of weights of unsatisfied clauses in which the variable x_i appears by the sum of weights of clauses connected to this variable.

Definition 3.2. *The global fitness C(S) is defined as the sum of the contribution from each variable, i.e.,*

$$C(S) = -\sum_{i=1}^{n} \left(\lambda_i \sum\nolimits_{x_i \in C_k} w_k \right) = -\sum_{i=1}^{n} (c_i \lambda_i), \text{ where } c_i = \sum\nolimits_{x_i \in C_k} w_k \tag{9}$$

where c_i is a constant for a given problem.

Remark 3.1. *The global fitness C(S) is a linear combination of the local fitness λ_i, which is consistent with the observation concerning fitness definition [42].*

Definition 3.3. *For the MAX-SAT problem, the exact backbone* $\mathbf{B}$ *is the set of variables having the same assignments in the set* $\mathbf{S}_{global}$ *of global optimal solutions. The formal definition is given as follows:*

$$\mathbf{B} = \{x_i \,|\forall s_j, s_k \in \mathbf{S}_{global}\,, s_j(x_i) = s_k(x_i)\} \tag{10}$$

Nevertheless, the exact backbone information of a given problem instance is even more difficult to obtain than actual problem solutions [30]. An approximate approach to estimating the backbone information is considering the local minima as "real" optimal ones.

Definition 3.4. *The quasi-backbone* $\mathbf{X_B}$ *is the set of variables having the same assignments in the set* $\boldsymbol{S}_{local}$ *of some local optimal solutions. Its formal definition is given as follows:*

$$\mathbf{X_B} = \{x_i \,|\forall s_j, s_k \in \mathbf{S}_{local}\,, s_j(x_i) = s_k(x_i)\} \tag{11}$$

In this paper, we incorporate the quasi-backbone information into the EO algorithm, and obtain the framework of "Backbone Guided Extremal Optimization (BGEO)", which is described in next page.

The BGEO framework can be viewed as an iterative process, which consists of backbone estimation phase and backbone-guided optimization phase. In the first iteration, i.e., $l = 1$, the BGEO collects R_l local optimum solutions starting from pure randomly generated initial solutions without any backbone information. In the following iterations, BGEO explores the complex landscape by utilizing the backbone information obtained in the last iteration. When the evolutionary probability distribution $P_l(k_1)$ adopted in BGEO algorithm is chosen as power-law, exponential, and hybrid distribution, respectively, the corresponding algorithm is called BG-PEO, BG-EEO, and BG-HEO, respectively, so the corresponding parameter p_l is τ_l, μ_l, and h_l, respectively.

Obviously, the performance of BGEO depends on these parameters including *MI*, R_l, SS_l, MS_l and p_l. Therefore, determining the appropriate values of these parameters to make BGEO achieve the best performance is a critical issue. Here, *MI* and R_l are all positive constants. According to the study on BE-EO [14], the parameters SS_l, MS_l are as follows:

$$SS_l = C_{l1} \times |\mathbf{X_{NB}}(l)| \tag{12}$$

$$MS_l = C_{l2} \times |\mathbf{X_{NB}}(l)| \tag{13}$$

where C_{l1}, C_{l2} are all positive constants and $|\mathbf{X_{NB}}(l)|$ is the number of non-backbone variables in the l-th iteration.

Now we focus on the adjustable parameter p_l for controlling the evolutionary probability distribution of BGEO. Obviously, p_l plays an analogous role to the proportion p of random and greedy moves in WALKSAT [34], and the noise parameter η in FMS [43]. Due to

Backbone Guided Extremal Optimization

Input: a MAX-SAT instance; *MI*: the maximum iterations; R_l: the maximum independent runs of the l-th iteration; SS_l: the maximum sample size in the l-th iteration; MS_l: the maximum steps of EO algorithm in the l-th iteration; p_l: the adjustable parameter for evolutionary probability distribution of EO algorithm in the l-th iteration;

Output: S_B: the best configurations found; $C(S_B)$: the total weights of unsatisfied clauses.

1: Initialization: set the backbone set $\mathbf{X_B} = \mathbf{0}$, non-backbone set $\mathbf{X_{NB}} = \mathbf{X}$,
2: **for** $l = 1$: MI
3: **for** $j = 1$: R_k
4: **for** $i = 1$: SS_k
5: Fix the values of $\mathbf{X_B}$, initialize $\mathbf{X_{NB}} = \mathbf{X} - \mathbf{X_B}$ by BEICG, and construct the initial solution S_i, set $S_{best} = S_i$
6: For the current solution S_i
(a). Evaluate λ_i for each variable x_i and rank all the variables according to λ_i, i.e., find a permutation Π_1 of the labels i such that $\lambda_{\Pi_1(1)} \geq \lambda_{\Pi_1(2)} \geq \cdots \geq \lambda_{\Pi_1(n)}$;
(b). Select a rank $\Pi(k)$ according to a probability distribution $P_l(k_1)$, $1 \leq k_1 \leq n$ and denote the corresponding variable as x_j;
(c). Flip the value of x_j and set $S_{new} = S_i$ in which x_j value is flipped;
(d). Accept $S_i = S_{new}$ unconditionally;
(e). If $C(S_{new}) \leq C(S_{best})$, then $S_{best} = S_{new}$;
7: Repeat the step 6 until the maximum steps MS_l, and obtain the best solution $S_{if} = S_{best}$
8: **end for**
9: Choose the best solution S_{bj} from the solution set $\{S_{if}\}$
10: **end for**
11: Obtain the solution set $\mathbf{S}_l = \{S_{bj}\}$, extract the backbone information from $\mathbf{S}_l$, update $\mathbf{X_B}$ and $\mathbf{X_{NB}}$
12: **end for**
13: Choose the best solution S_B from $\mathbf{S} = \bigcup_{l=1}^{MI} \mathbf{S}_l$, and obtain the corresponding cost $C(S_B)$

the different features of the first and the remaining iterations, p_l is given in the following form:

$$p_l = \begin{cases} p_c, & l = 1 \\ p_c + d * |\mathbf{X_B}(l-1)|, & 2 \leq l \leq MI \end{cases} \tag{14}$$

where p_c is the initial value of the parameter p_l and d is a positive constant.

From the BGEO framework, it is clear that the optimization of the $(l+1)$-th iteration always starts from the initial solutions where are all based on the backbone information extracted in l-th iteration, so the size of the backbone extracted in $(l+1)$-th iteration is generally more than at least equal to that in l-th iteration, i.e., $n \geq |\mathbf{X_B}(MI)| \geq \cdots \geq |\mathbf{X_B}(l+1)| \geq |\mathbf{X_B}(l)| \geq \cdots \geq |\mathbf{X_B}(l)|$. In other words, the size of the remaining problem that needs to be optimized will be smaller and smaller as the number of iteration increases. As a consequence, there must exist a finite constant $MI_{\max}$ such that $|\mathbf{X_B}(MI_{\max})| \to n$. To illustrate this observation, Figure 1 shows the dynamics of the pseduo backbone size during the search process of BGEO for some uniform satisfiable MAX-3-SAT instances

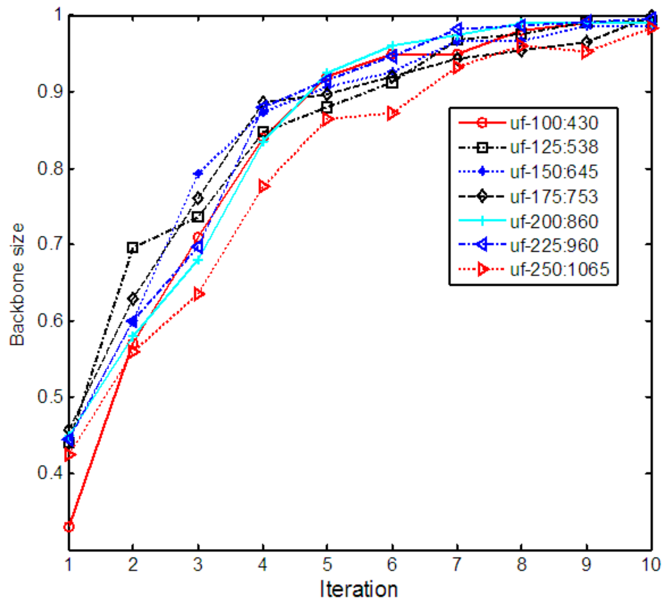

FIGURE 1. The dynamics of the preduo backbone size during the search process of BGEO

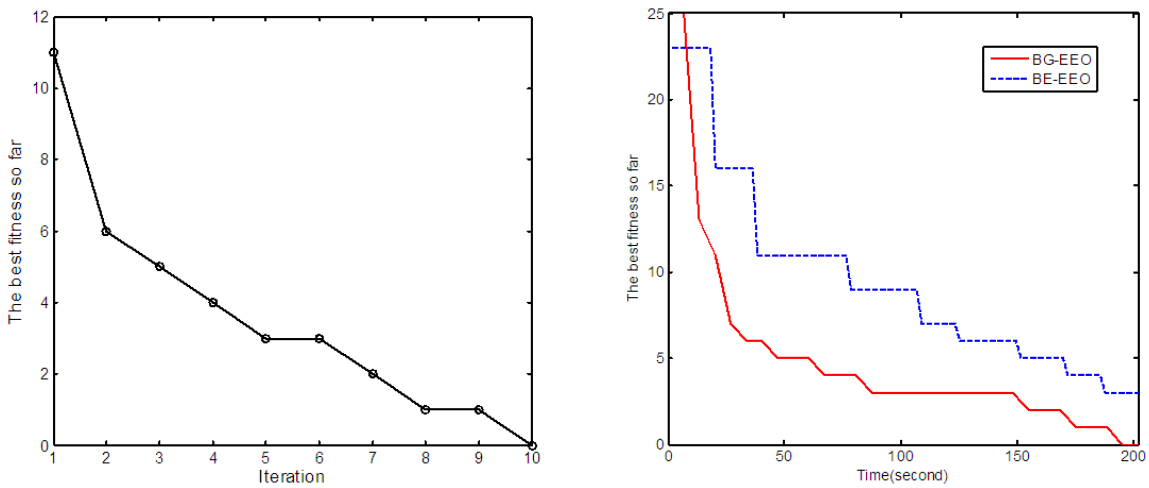

FIGURE 2. For uf-100:430, the left is the search dynamics of the best global fitness in BGEO and the right is the comparison of BG-EEO and BE-EEO

"uf-n:m" [40], in which n is the number of the variables and m is the number of the clauses. The backbone size increases as the number of iteration increases.

For the uf-100:430 instance, the search dynamics of the best global fitness in BGEO are shown in the left of Figure 2. For the hard MAX-3-SAT instances, BGEO can reach high-quality solutions in finite iterations. The right of Figure 2 gives the performance comparison of BE-EEO and BGEO algorithms. Obviously, BGEO performs better than BE-EEO in the same runtime. Furthermore, the superiority of BGEO algorithm is demonstrated by the experimental results in next section.

4. **Experimental Results and Discussion.** In order to demonstrate the effectiveness of the BGEO, we choose the hard MAX-SAT problem instances from SATLIB [44] as a testbed. The tested problems include random unweighted MAX-3-SAT instances, MAX-3-SAT instances near phase transition, and MAX-3-SAT instances with controlled backbone size. Note that all algorithms are implemented in MATLIB7.6 on a Pentium 1.86 GHz PC with dual-core processor T2390 and 2GB RAM running Windows Vista Basic systems. The performances of these algorithms are measured by the best, mean, and worst errors denoted as e_b, e_m and e_w respectively. The errors are defined as $e_b(\%) = 100 \times (m_b - m_o)/m$, $e_m(\%) = 100 \times (m_m - m_o)/m$ and $e_w(\%) = 100 \times (m_w - m_o)/m$ respectively, where m_b, m_m and m_w are the minimal, average and maximal number of unsatisfied clauses over 10 independent runs respectively and m_o is the optimal solution.

Remark 4.1. *The experimental results [14] on random and structured MAX-SAT instances have shown that BE-EO can provide better or at least competitive performance than more elaborate stochastic optimization methods, such as SA [33], GSAT [34], WSAT [35], and TS-CSP [36]. Furthermore, the superiority of the BE-EEO and BE-HEO algorithms under EOSAT framework to the BE-EO algorithm is demonstrated by our recent research [37]. Consequently, this paper concentrates on comparing BG-EEO with these reported algorithms by the experiments on the random unweighted MAX-3-SAT instances [14].*

The results on the random unweighted MAX-3-SAT instances are shown as Table 1, where the performances of these algorithms are measured by the average error (%). It is obvious that BG-EEO is superior to these reported algorithms.

Table 1. Comparsion of BG-EEO and the reported algorithms on random unweigheted MAX-3-SAT instances

Variables (n)	100	100	300	300	500
Clauses (m)	500	700	1500	2000	5000
SA [33]	1.64	2.587	2.000	2.900	4.528
TS-CSP [36]	0.453	1.755	0.523	1.595	3.328
GWSAT [34]	0.556	1.914	0.551	1.597	3.279
WSAT [35]	0.552	1.914	0.541	1.614	3.340
τ-EO [14]	0.800	1.880	0.600	1.900	3.352
BE-EO [14]	0.632	1.860	0.500	1.550	3.100
BE-EEO [37]	0.524	1.732	0.486	1.479	3.003
BG-EEO	0.400	1.553	0.264	1.245	2.875

Remark 4.2. *The tested MAX-3-SAT satisfiable (unsatisfiable) instances are represented as "uf-n:m (uuf-n:m)" here, in which n is the number of the variables and m is the number of the clauses. For example, 'uuf-50:218' represents the unsatisfiable instance has 50 variables and 218 clauses. These instances with* $\alpha = m/n$ *ranges from 4.260 to 4.360, which are close to the critical threshold of phase transition* $\alpha_c \approx 4.267$. *For each unsatisfiable instance, the optimal number of unsatisfied clauses is 1, i.e.,* $m_o = 1$. *Therefore, we focus on the optimization problem, MAX-3-SAT, that is to find an assignment to maximize the number of satisfied clauses. In other words, MAX-3-SAT is equivalent to minimize the number of unsatisfied clauses.*

The experimental results on these satisfiable and unsatisfiable instances near phase transition are shown in Table 2 and Table 3, respectively. Table 4 gives the comparison

Table 2. Comparsion of BG-EEO and the reported BE-EO [14], BE-EEO [37] algorithms for MAX-3-SAT satisfiable instances near phase transition

Problem	α	BE-EO [10]			BE-EEO [33]			BG-EEO		
		e_b	e_m	e_w	e_b	e_m	e_w	e_b	e_m	e_w
uf-50:218	4.360	0.92	2.20	2.75	0.46	1.74	2.29	0.00	0.00	0.00
uf-75:325	4.333	1.54	2.62	3.07	0.62	1.45	2.15	0.00	0.15	0.31
uf-100:430	4.300	1.63	2.47	2.80	0.70	1.83	2.33	0.00	0.23	0.46
uf-125:538	4.304	2.04	2.68	3.35	1.11	1.99	2.23	0.19	0.28	0.37
uf-150:645	4.300	2.17	2.53	2.95	1.40	1.95	2.33	0.16	0.31	0.47
uf-175:753	4.303	2.39	2.76	2.92	1.59	1.91	2.26	0.13	0.33	0.40
uf-200:860	4.300	2.67	3.13	3.49	1.86	2.29	2.56	0.12	0.17	0.23
uf-225:960	4.267	2.08	2.79	3.23	1.67	1.99	2.29	0.10	0.31	0.62
uf-250:1065	4.260	1.88	2.76	3.29	1.60	1.94	2.07	0.09	0.35	0.66

Table 3. Comparsion of BG-EEO and the reported BE-EO [14], BE-EEO [37] algorithms for MAX-3-SAT unsatisfiable instances near phase transition

Problem	α	BE-EO [10]			BE-EEO [33]			BG-EEO		
		e_b	e_m	e_w	e_b	e_m	e_w	e_b	e_m	e_w
uuf-50:218	4.360	1.38	2.38	3.21	0.46	1.88	2.75	0.00	0.00	0.00
uuf-75:325	4.333	1.85	2.65	3.37	1.23	1.94	2.46	0.00	0.17	0.31
uuf-100:430	4.300	1.86	2.63	2.80	1.16	1.88	2.56	0.00	0.26	0.46
uuf-125:538	4.304	1.86	2.70	3.35	1.30	2.08	3.16	0.19	0.30	0.37
uuf-150:645	4.300	2.17	2.71	3.41	1.40	2.05	2.64	0.16	0.32	0.47
uuf-175:753	4.303	2.92	3.33	3.98	1.73	2.30	2.67	0.13	0.34	0.40
uuf-200:860	4.300	3.49	3.85	4.30	2.44	2.72	3.14	0.12	0.16	0.23
uuf-225:960	4.267	2.81	3.48	4.17	2.19	2.64	3.44	0.10	0.44	0.62
uuf-250:1065	4.260	3.09	3.51	4.38	1.78	2.28	2.72	0.09	0.45	0.66

Table 4. BG-EEO vs. BE-EEO algorithm [37] for the large MAX-SAT instances

Problems	α	BE-EEO [33]			BG-EEO		
		e_b	e_m	e_w	e_b	e_m	e_w
f600_2550	4.25	1.73	2.96	3.89	0.52	0.96	1.57
f1000_4250	4.25	2.08	3.35	4.74	0.67	1.21	1.92
f2000_8500	4.25	2.87	4.01	4.92	0.83	1.50	2.33

of BG-EEO and BE-EEO for some large instances. Clearly, BG-EEO outperforms the reported BE-EO [14] and BE-EEO [37] algorithms for these hard instances. Especially, the BG-EEO algorithm reaches the optimal solutions for some instances shown as the bold.

Remark 4.3. *MAX-3-SAT instances with controlled backbone size (CBS) are different from those given in the next subsection in that they have some backbone variables, where the backbone size is defined as b. Singer et al. [22] have shown that the search cost is very high even for the small size problems but with large backbone size. Therefore, these CBS instances from SATLIB are chosen for testing the superiority of the proposed BGEO method. The control parameter α of these instances ranging from 4.03 to 4.49 is near the critical threshold of the phase transition $\alpha_c \approx 4.267$. Moreover, the values of b in these instances range from 10 to 90 at each α value.*

TABLE 5. BG-EEO vs. the BE-EO [14], BE-EEO [37] algorithms for CBS instances

Problem	α	b	BE-EO [10]			BE-EEO [33]			BG-EEO		
			e_b	e_m	e_w	e_b	e_m	e_w	e_b	e_m	e_w
CBS_100_403	4.03	10	1.99	2.68	3.23	0.99	1.91	2.48	**0.00**	0.07	0.25
		30	1.99	2.46	2.98	1.24	1.76	2.23	0.25	0.45	0.74
		50	1.49	2.06	2.48	1.24	1.61	1.98	0.50	0.74	0.99
		70	2.43	2.75	3.16	1.24	1.74	1.98	0.50	0.74	0.99
		90	2.23	2.73	3.47	1.74	2.13	2.48	0.74	0.99	1.24
CBS_100_411	4.11	10	1.46	2.12	2.68	0.97	1.58	1.95	0.24	0.49	0.73
		30	1.22	2.19	2.92	1.22	1.68	2.68	0.24	0.49	0.97
		50	1.95	2.68	3.16	1.70	2.09	2.92	0.24	0.61	0.97
		70	1.95	2.70	3.89	1.70	2.12	2.19	0.49	0.78	0.97
		90	1.95	2.53	3.41	1.46	1.85	2.19	0.24	0.68	0.97
CBS_100_418	4.18	10	1.67	2.34	3.11	1.22	1.75	2.19	0.24	0.38	0.72
		30	2.39	2.68	3.11	1.44	2.08	2.39	0.24	0.36	0.72
		50	2.63	2.73	3.11	1.44	2.03	2.39	0.48	0.72	0.96
		70	0.96	2.44	3.11	0.96	1.48	2.15	0.48	0.62	0.72
		90	2.15	2.68	3.11	1.20	1.87	2.39	0.48	0.72	0.96
CBS_100_423	4.23	10	1.65	2.13	2.60	0.71	1.47	2.13	**0.00**	0.47	0.95
		30	1.42	2.48	3.55	1.18	1.84	2.13	0.24	0.47	0.71
		50	1.65	2.53	3.07	1.18	1.80	2.36	0.24	0.57	0.71
		70	1.42	2.41	2.84	0.95	1.77	2.36	0.24	0.54	0.71
		90	1.89	2.55	3.31	1.42	2.00	2.13	0.24	0.80	0.95
CBS_100_429	4.29	10	1.17	2.45	3.03	0.47	1.52	2.10	**0.00**	0.65	0.93
		30	1.63	2.17	2.56	1.17	1.75	2.10	**0.00**	0.47	0.70
		50	2.10	2.66	3.26	0.93	1.70	2.33	**0.00**	0.47	0.70
		70	1.40	2.24	2.80	0.70	1.70	2.33	0.23	0.58	0.93
		90	2.10	2.77	3.26	1.86	2.24	2.56	0.23	0.70	0.93
CBS_100_435	4.35	10	1.15	1.91	2.53	0.69	1.22	1.61	**0.00**	0.28	0.46
		30	1.38	1.91	2.53	0.92	1.47	1.84	**0.00**	0.32	0.46
		50	2.30	2.67	3.45	1.61	2.05	2.30	0.46	0.69	0.92
		70	1.61	2.46	2.99	1.15	1.89	2.30	0.23	0.64	0.92
		90	2.30	2.57	2.76	0.92	1.66	2.30	0.23	0.60	0.92
CBS_100_441	4.41	10	1.36	2.12	2.49	0.68	1.45	2.04	0.23	0.45	0.68
		30	1.59	2.15	2.95	0.45	1.16	2.04	**0.00**	0.23	0.45
		50	1.81	2.59	3.40	1.13	1.88	2.72	0.23	0.45	0.91
		70	1.81	2.49	2.95	1.59	1.90	2.49	0.45	0.68	0.91
		90	2.49	2.90	3.63	1.36	2.06	2.72	0.45	0.73	0.91
CBS_100_449	4.49	10	1.56	2.27	3.12	0.67	1.44	2.00	0.22	0.45	0.67
		30	1.78	2.45	3.12	1.34	1.87	2.23	0.45	0.58	0.89
		50	1.78	2.05	2.90	0.89	1.27	1.78	0.22	0.33	0.67
		70	2.45	2.92	3.34	1.34	2.16	2.67	0.22	0.56	0.89
		90	2.67	3.27	3.79	1.56	2.32	2.90	0.67	0.89	1.11

The comparison of BG-EEO and the reported BE-EO and BE-EEO algorithms for these CBS instances is shown in Table 5. It is clear that BG-EEO performs much better than BE-EO [14] and BE-EEO [37] for these hard CBS instances. Especially, BG-EEO algorithm reaches the optimal solutions for some instances shown as the bold.

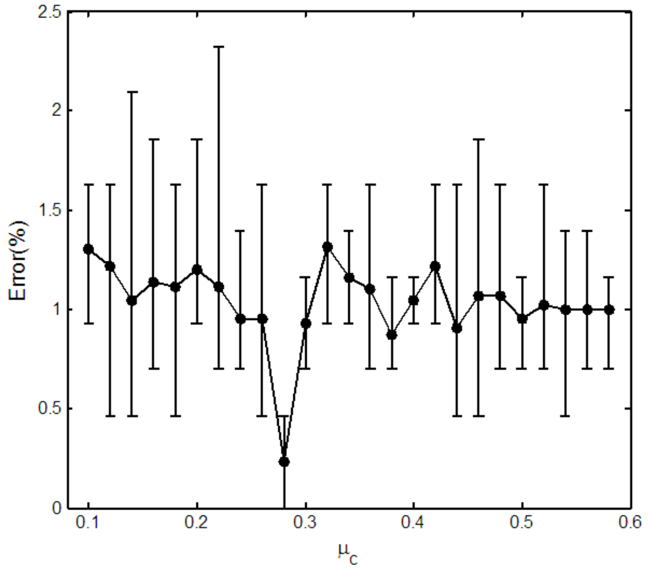

FIGURE 3. The control parameter μ_c vs. the performance of BG-EEO

As analyzed in previous section, the parameters in BGEO, such as MI, R_l, SS_l, MS_l, are determined easily. For the above experiments, these parameters MI, R_l are set as $MI = 10$, R_l=3 and SS_l, MS_l are determined according to Equations (12) and (13), respectively. The evolutionary probability distribution control parameter p_c plays a critical role in governing the performance of BGEO. The optimal values of p_c are determined according to Equation (14). Figure 3 illustrates the effects of μ_c on the performance of BG-EEO for the uuf-100:430 instance. Obviously, the optimal value of μ_c is approximately 0.28 for the uuf-100:430 instance. Similarly, for other MAX-3-SAT instances in Tables 1 and 2, the optimal values of μ_c are approximately from 0.24 to 0.36. For the instances in Table 3, the optimal values are approximately from 0.26 to 0.32.

5. **Conclusions.** In this paper, we develop a novel optimization method called 'backbone guided extremal optimization (BGEO)' for the hard SAT and MAX-SAT problems. The basic idea behind BGEO is to incorporate the backbone information extracted from the history of search process into EO to guide the entire search process approach the optimal or at least high-quality solutions. The BGEO is essentially a biased local search method that exploits the "big valley" structure of the configuration space [30]. Also, it is similar to the population learning the large-scale structure of the fitness landscape [45]. The experimental results on a variety of hard MAX-SAT problem instances have shown that BGEO outperforms the reported BE-EO algorithm. Nevertheless, it should be stressed that the main purpose of this research is to develop a new optimization method rather than provide the best algorithm for a particular problem. As a consequence, the performance of BGEO is possible to be further improved by fine-tuning the control parameters or adapting other backbone-guided search strategies, which is well-studied in our future work. In fact, how to design an effective adaptive search strategy under the BGEO framework is future work.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (No. 61074045), the Zhejiang Provincial Natural Science Foundation of China (No. Y6090220) and the Open Project Program (No. ICT1112) from State Key Laboratory of Industrial Control Technology (Zhejiang University). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] S. Boettcher and A. G. Percus, Nature's way of optimizing, *Artif. Intell.*, vol.119, pp.275-286, 2000.
[2] S. Boettcher and A. G. Percus, Optimization with extremal dynamics, *Phys. Rev. Lett.*, vol.86, no.23, pp.5211-5214, 2001.
[3] P. Bak and K. Sneppen, Punctuated equilibrium and criticality in a simple model of evolution, *Phys. Rev. Lett.*, vol.71, no.24, pp.4083-4086, 1993.
[4] P. Bak, C. Tang and K. Wiesenfeld, Self-organized criticality, *Phys. Rev. Lett.*, vol.59, pp.381-384, 1987.
[5] Z. Cui, X. Cai and J. Zeng, Chaotic performance-dependent particle swarm optimization, *International Journal of Innovative Computing, Information and Control*, vol.5, no.4, pp.951-960, 2009.
[6] S. Boettcher and A. G. Percus, Extremal optimization for graph partitioning, *Phys. Rev. E*, vol.64, no.2, 2001.
[7] S. Boettcher and A. G. Percus, Extremal optimization at the phase transition for the three-coloring problem, *Phys. Rev. E*, vol.69, 2004.
[8] A. A. Middleton, Improved extremal optimization for the Ising spin glass, *Phys. Rev. E*, vol.69, 2004.
[9] K. Hamacher, Adaptive extremal optimization by detrended fluctuation analysis, *J. Comput. Phys.*, vol.227, pp.1500-1509, 2007.
[10] T. Zhou, W. J. Bai, L. L. Cheng, and B. H. Wang, Continuous extremal optimization for Lennard-Jones clusters, *Phys. Rev. E*, vol.72, no.1, 2005.
[11] Y. W. Chen, Y. Z. Lu and P. Chen, Optimization with extremal dynamics for the travelling salesman problem, *Physica A*, vol.385, pp.115-123, 2007.
[12] J. M. Liu, Y. W. Chen, G. K. Yang and Y. Z. Lu, Self-organized combinatorial optimization, *Expert Systems with Applications*, vol.38, pp.10532-10540, 2011.
[13] G. Q. Zeng, Y. Z. Lu, W. J. Mao and J. Chu, Study on probability distributions for evolution in modified extremal optimization, *Physica A*, vol.389, pp.1922-1930, 2010.
[14] M. B. Menaï and M. Batouche, An effective heuristic algorithm for the maximum satisfiability problem, *Appl. Intell.*, vol.24, pp.227-239, 2006.
[15] F. L. Sousa, F. M. Ramos and P. Paglione, New stochastic algorithm for design optimization, *American Institute of Aeronautics and Astronautics*, vol.41, no.9, pp.1808-1818, 2003.
[16] F. L. Sousa, V. Vlassov and F. M. Ramos, Generalized extremal optimization: An application in heat pipe design, *Applied Mathematical Modelling*, vol.28, pp.911-931, 2004.
[17] Y. W. Chen, Y. Z. Lu and G. K. Yang, Hybrid evolutionary algorithm with marriage of genetic algorithm and extremal optimization for production scheduling, *International Journal of Advanced Manufacturing Technology*, vol.36, no.9-10, pp.959-968, 2008.
[18] S. Boettcher, Evolutionary dynamics of extremal optimization, *Lecture Notes on Computer Science*, vol.5581, pp.1-14, 2009.
[19] G. Q. Zeng and Y. Z. Lu, Survey on computational complexity with phase transitions and extremal optimization, *Proc. of the 48th IEEE conf. Control and Decision & the 28th Chinese Control Conf.*, pp.4352-4359, 2009.
[20] J. Slaney and T. Walsh, Backbones in optimization and approximation, *Proc. of IJCAI*, pp.254-259, 2001.
[21] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman and L. Troyansky, Determining computational complexity from characteristic 'phase transition', *Nature*, vol.400, pp.133-137, 1999.
[22] J. Singer, I. P. Gent and A. Smaill, Backbone fragility and the local search cost peak, *Journal of Artificial Intelligence Research*, vol.12, pp.235-270, 2000.
[23] W. X. Zhang, Phase transitions and backbones of 3-SAT and maximum 3-SAT, *Proc. of CP*, pp.153-167, 2001.
[24] W. X. Zhang, Phase transitions, backbones, measurement accuracy, and phase-aware approximation: The ATSP as a case study, *Proc. of CP-AI-OR*, pp.345-357, 2002.
[25] P. Kilby, J. Slaney, S. Thiébaux and T. Walsh, Backbones and backdoors in satisfiability, *Proc. of the 12th National Conference on Artificial Intelligence*, 2005.
[26] J. Schneider, C. Froschhammer, I. Morgenstern, T. Husslein and J. M. Singer, Searching for backbones-efficient parallel algorithm for the traveling salesman problem, *Comput. Phys. Commun.*, vol.96, pp.173-188, 1996.

[27] J. Schneider, Searching for backbones – A high-performance parallel algorithm for solving combinatorial optimization problem, *Future Generation Computer Systems*, vol.19, pp.121-131, 2003.
[28] O. Dubolis and G. Dequen, A backbone-search heuristic for efficient solving of hard 3-SAT formulae, *Proc. of International Joint Conference on Artificial Intelligence*, pp.248-253, 2001.
[29] O. Telelis and P. Stamatopoulos, Heuristic backbone sampling for maximum satisfability, *Proc. of the 2nd Hellenic Conference on Artificial Intelligence*, pp.129-139, 2002.
[30] W. X. Zhang, Configuration landscape analysis and backbone guided local search. Part I: Satisfiability and maximum satisfiability, *Artif. Intell.*, vol.158, pp.1-26, 2004.
[31] W. X. Zhang and M. Looks, A novel local search algorithm for the travelling salesman problem that exploits backbones, *Proc. of the 19th International Joint Conference on Artificial Intelligence*, pp.343-350, 2005.
[32] M. E. B. Menaï and M. Batouche, An backbone-based co-evolutionary heuristic for partial MAX-SAT, *Proc. of the 7th International Conference on Artificial Evolution*, pp.155-166, 2006.
[33] P. Hansen and B. Jaumard, Algorithms for maximum satisfiability problems, *Computing*, vol.44, pp.279-303, 1990.
[34] B. Selman and H. A. Kautz, An empirical study of greedy local search for satisfiability testing, *Proc. of the 11th National Conference on Aritificial Intelligence*, pp.46-51, 1993.
[35] B. Selman, H. A. Kautz and B. Cohen, Noise strategies for improving local search, *Proc. of the 12th National Conference on Artificial Intelligence*, pp.337-343, 1994.
[36] F. Glover, Tabu search: Part I, *ORSA J. Comp.*, vol.1, no.3, pp.190-206, 1989.
[37] G. Q. Zeng, Y. Z. Lu and W. J. Mao, Modified extremal optimization for the maximum satisfiability problem, *Journal of Zhejiang University SCIENCE C*, vol.12, no.7, pp.589-596, 2011.
[38] A. Biere, M. Heule, H. Maaren and T. Walsh, *Handbook of Satisfiability*, IOS Press, 2009.
[39] M. Mézard, G. Parisi and R. Zecchina, Analytic and algorithmic solution of random satisfiability problems, *Science*, vol.297, pp.812-815, 2002.
[40] H. Haken and H. C. Wolf, *The Physics of Atoms and Quanta*, 5th Edition, Springer-Verlag, 1996.
[41] S. Szedmak, How to find more efficient initial solution for searching, *RUTCOR Research Report 49-2001*, Rutgers Center for Operations Research, Rutgers University, Piscataway, NJ, USA, 2001.
[42] K. H. Hoffmann, F. Heilmann and P. Salamon, Fitness threshold accepting over extremal optimization ranks, *Phys. Rev. E*, vol.70, no.4, 2004.
[43] S. Seitz, M. Alava and P. Orponen, Focused local search for random 3-satisfiability, *J. Stat. Mech.: Theor. Exp.*, 2005.
[44] *SATLIB – Benchmark Problems*, http://www.cs.ubc.ca/∼hoos/SATLIB/benchm.html.
[45] M. Qasem and A. P. Bennett, Learning the large-scale structure of the MAX-SAT landscape using populations, *IEEE Transactions on Evolutionary Computation*, vol.14, no.4, pp.518-529, 2010.