Sporadic Model Building for Efficiency Enhancement of Hierarchical BOA

Martin Pelikan
Missouri Estimation of Distribution Algorithms Laboratory, 320 CCB
Dept. of Mathematics and Computer Science
Univ. of Missouri–St. Louis
One University Blvd.
St. Louis, MO 63121
pelikan@cs.umsl.edu

Kumara Sastry
Illinois Genetic Algorithms Laboratory, 107 TB
University of Illinois at Urbana-Champaign
104 S. Mathews Ave.
Urbana, IL 61801
kumara@illigal.ge.uiuc.edu

David E. Goldberg
Illinois Genetic Algorithms Laboratory, 107 TB
University of Illinois at Urbana-Champaign
104 S. Mathews Ave.
Urbana, IL 61801
deg@illigal.ge.uiuc.edu

ABSTRACT

This paper describes and analyzes *sporadic model building*, which can be used to enhance the efficiency of the hierarchical Bayesian optimization algorithm (hBOA) and other advanced estimation of distribution algorithms (EDAs) that use complex multivariate probabilistic models. With sporadic model building, the structure of the probabilistic model is updated once every few iterations (generations), whereas in the remaining iterations only model parameters (conditional and marginal probabilities) are updated. Since the time complexity of updating model parameters is much lower than the time complexity of learning the model structure, sporadic model building decreases the overall time complexity of model building. The paper shows that for boundedly difficult nearly decomposable and hierarchical optimization problems, sporadic model building leads to a significant *model-building speedup* that decreases the asymptotic time complexity of model building in hBOA by a factor of $\Theta(n^{0.26})$ to $\Theta(n^{0.5})$, where n is the problem size. On the other hand, sporadic model building also increases the number of evaluations until convergence; nonetheless, the *evaluation slowdown* is insignificant compared to the gains in the asymptotic complexity of model building.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search;
I.2.6 [**Artificial Intelligence**]: Learning;
G.1.6 [**Numerical Analysis**]: Optimization

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

General Terms

Algorithms

Keywords

Bayesian optimization algorithm, hierarchical BOA, estimation of distribution algorithms, efficiency enhancement, sporadic model building.

1. INTRODUCTION

The hierarchical Bayesian optimization algorithm (hBOA) [19, 20, 18] replaces standard variation operators of genetic and evolutionary algorithms [10, 5, 24, 12] by building a Bayesian network for selected solutions and sampling the built network to generate new candidate solutions. Additionally, hBOA uses restricted tournament replacement [8] to effectively maintain diversity and preserve alternative partial solutions. It was theoretically and empirically shown that hBOA can solve nearly decomposable and hierarchical optimization problems in a quadratic number of evaluations or faster [18, 23]. However, quadratic or subquadratic performance may be still insufficient for problems with thousands of decision variables or high-order interactions. Consequently, efficiency enhancement techniques [6, 25, 27, 18] may have to be incorporated into hBOA to make this algorithm practical even for extremely large and complex problems.

A number of efficiency enhancement techniques can be incorporated into hBOA and other genetic and evolutionary algorithms [6, 27, 1, 28, 29, 18]. This paper discusses *sporadic model building*, which can significantly speed up model building in hBOA and other advanced estimation of distribution algorithms (EDAs) that use complex multivariate probabilistic models. Specifically, with sporadic model building, hBOA updates the structure of the Bayesian network used to sample new solutions once every few iterations (generations); in the remaining iterations, the structure from the previous iteration is used and only the parameters of the network (conditional and marginal probabilities) are updated based on the selected solutions. Since learning the structure is the most expensive component of model building, sporadic model building should lead to a significant speedup of model building. This paper describes and analyzes sporadic model building in hBOA. The results indicate that sporadic model building leads to a significant speedup of model building that decreases the asymptotic complexity of model building in hBOA.

The paper starts by describing hBOA and discussing its time complexity in Section 2. Section 3 presents sporadic model building and discusses its effects on the time complexity of hBOA. Section 4 presents and discusses experimental results. Section 5 discusses future work in this area. Finally, Section 6 summarizes and concludes the paper.

2. HIERARCHICAL BAYESIAN OPTIMIZATION ALGORITHM (HBOA)

Estimation of distribution algorithms (EDAs) [14, 22, 13] evolve a population of candidate solutions to the given problem by building and sampling a probabilistic model of promising solutions. The hierarchical Bayesian optimization algorithm (hBOA) [19, 20, 18] is an EDA that uses Bayesian networks to represent the probabilistic model and incorporates restricted tournament replacement for effective diversity maintenance. This section outlines hBOA and discusses time complexity of the methods for learning and sampling Bayesian networks used to guide exploration of the search space in hBOA.

2.1 Basic hBOA Procedure

hBOA evolves a population of candidate solutions represented by fixed-length strings over a finite alphabet (for example, binary strings). The initial population is generated at random according to a uniform distribution over the set of all potential solutions. Each iteration (generation) starts by selecting promising solutions from the current population using any standard selection method of genetic and evolutionary algorithms. In this paper we use binary tournament selection without replacement.

After selecting the promising solutions, hBOA builds a Bayesian network [11, 16] with local structures as a model for these solutions. New solutions are generated by sampling the built network. The new solutions are then incorporated into the original population using restricted tournament replacement (RTR) [8], which ensures effective diversity maintenance. RTR with window size $w > 1$ incorporates each new candidate solution X into the original population using the following three steps: (1) Randomly select a subset W of w candidate solutions from the original population. (2) Let Y be a solution from W that is most similar to X (based on genotypic distance). (3) Replace Y with X if X is better; otherwise, discard X. A robust rule of thumb is to set $w = \min\{n, N/20\}$, where n is the number of decision variables in the problem and N is the population size [18].

The next iteration is executed unless some predefined termination criteria are met. For example, the run can be terminated when the maximum number of generations is reached or the entire population consists of copies of the same candidate solution. For more details about the basic hBOA procedure, see [19] or [18].

It is beyond the scope of this paper to provide a detailed description of hBOA variation, which consists of learning and sampling Bayesian networks with decision trees; for more information, please see [3], [9], and [18]. Nonetheless, before discussing sporadic model building, it is important to understand the time complexity of the different components of hBOA variation and that is why the remainder of this section discusses the time complexity of learning and sampling Bayesian networks with decision trees.

2.2 Time Complexity of Learning and Sampling Bayesian Networks

Learning a Bayesian network with local structures consists of two steps [9]: (1) learn the structure, and (2) learn the parameters (conditional probabilities).

Assuming that the problem is decomposable into subproblems of order k, the time complexity of learning the structure of a BN with decision trees can be bounded by $O(kn^2N)$ [18]. Under the same assumptions, the conditional probabilities can be computed in $O(knN)$ time steps. Thus, the asymptotic time complexity of learning the network parameters is *much lower* than the asymptotic time complexity of learning the network structure.

Assuming an order-k decomposable problem, sampling one candidate solution from a given BN can be done in $O(kn)$ steps. Consequently, the asymptotic complexity of sampling a population of N candidate solutions can be bounded by $O(knN)$.

Therefore, the asymptotic time complexity of building the network structure dominates the overall complexity of the variation operator that consists of building and sampling a BN. Similar behavior can be observed in other EDAs that use complex probabilistic models, for example, in the extended compact genetic algorithm (ECGA) [7, 26]. The following section describes sporadic model building, which can be used to speedup hBOA and other advanced EDAs.

3. SPORADIC MODEL BUILDING (SMB)

The above section indicated that building the network structure is the most expensive part of hBOA variation. Consequently, for large problems, building the network structure may become a bottleneck. To speed up the structure-building procedure, one can exploit the fact that although the model structure does not remain the same throughout the run, the structures in consequent iterations of hBOA are usually very similar. *Sporadic model building* (SMB) exploits this behavior by building the structure once every few iterations, while in the remaining iterations only the parameters are updated for the structure used in the previous iteration.

The remainder of this section discusses a simple schedule that can be used to control SMB; next, it discusses the effects of SMB on the time complexity of hBOA.

3.1 Simple Periodic Schedule to Control SMB

One of the most important factors that influences the effectiveness of SMB is the *schedule* that determines when to build the structure and when to only update the parameters for the previous structure. This paper investigates a simple approach that uses a parameter t_{sb} called the structure-building period, which denotes the period with which the structure is updated. For example, $t_{sb} = 1$ denotes the scenario in which the structure is built in every iteration, whereas $t_{sb} = 2$ denotes the scenario in which the structure is built only in every other iteration. Of course, other types of schedules can be used; nonetheless, this paper argues that even a simple schedule based on a fixed period yields asymptotic speedups of model building in hBOA and similar results can be expected in other advanced EDAs.

3.2 Effects of SMB on BOA Time Complexity

There are two potential effects of using SMB in BOA: speedup of model building and slowdown of evaluation:

1. *Speedup of model building.* Since with SMB, the most expensive part of the model building procedure is run only each t_{sb}th iteration, increasing t_{sb} should lead to a speedup of model building. Ideally, the speedup of building the model structure would be linearly proportional to t_{sb}; nonetheless, SMB may lead to an increase of the population size and the number of iterations and, as a result, with SMB each model building step may become more computationally expensive than without SMB. Furthermore, the actual speedup for a fixed problem size must be upper bounded by the expected number of iterations until convergence (the structure must be learned at least once).

   Since the number of iterations should be bounded by $O(n)$ where n is the number of bits [15, 31], the speedup of model building is expected to be bounded by $O(n)$. Another way to estimate an upper bound of the expected speedup is to argue that the asymptotic complexities of building the structure and learning the parameters differ by a factor of $\Theta(n)$ and no matter whether the structure is built or not, the parameters must be updated in each iteration.

2. *Slowdown of evaluation.* SMB may lead to a decreased accuracy of the probabilistic models used to sample new candidate solutions. As a result, the population size N for building a sufficiently accurate model may increase with t_{sb}. Additionally, because of a less frequent adaptation of the model structure to the current population of promising solutions, SMB may slow down the convergence and the total number G of iterations may be expected to increase with t_{sb}.

   Consequently, an increase in the overall time spent in the evaluation of candidate solutions can be expected as a result of the increase of N and G, because the time spent in evaluation is linearly proportional to the number E of evaluations, where $E = N \times G$.

It can be expected that for each particular problem, there exists an optimal value of t_{sb}, which leads to the maximum overall speedup of hBOA on this problem. Nonetheless, to make our results independent of the time complexity of the evaluation function, which changes from problem to problem, we studied the above two effects in separation. The following section provides empirical results that approximate the two effects for several nearly decomposable and hierarchical problems.

4. EXPERIMENTS

This section presents experimental results. First, test problems are described and the experimental methodology is discussed. Next, the results are presented and discussed.

4.1 Test Problems

This section describes test problems used to test sporadic model building in hBOA. All test problems are nearly decomposable or hierarchical and assume that candidate solutions are represented by n-bit binary strings. Three test problems were used: (1) dec-3, (2) trap-5, and (3) hTrap:

1. *Dec-3: Concatenated 3-bit deceptive function.* In dec-3 [4], the input string is first partitioned into independent groups of 3 bits each. This partitioning is unknown to the algorithm and it does not change during the run. A 3-bit deceptive function is applied to each group of 3 bits and the contributions of all deceptive functions are added together to form the fitness. Each 3-bit deceptive function is defined as follows:

   $$dec(u) = \begin{cases} 1 & \text{if } u = 3 \\ 0 & \text{if } u = 2 \\ 0.8 & \text{if } u = 1 \\ 0.9 & \text{if } u = 0 \end{cases}, \qquad (1)$$

   where u is the number of 1s in the input string of 3 bits. The task is to maximize the function. An n-bit dec-3 function has one global optimum in the string of all 1s and $(2^{n/3} - 1)$ other local optima. To solve dec-3, it is necessary to consider interactions among the positions in each partition because when each bit is considered independently, the optimization is misled away from the optimum [30, 2, 21].

2. *Trap-5: Concatenated 5-bit trap.* Trap-5 is defined analogically to dec-3, but instead of 3-bit groups, 5-bit groups are considered. The contribution of each group of 5 bits is computed as

   $$trap_5(u) = \begin{cases} 5 & \text{if } u = 5 \\ 4 - u & \text{otherwise} \end{cases}, \qquad (2)$$

   where u is the number of 1s in the input string of 5 bits. The task is to maximize the function. An n-bit trap5 function has one global optimum in the string of all 1s and $(2^{n/5}-1)$ other local optima. Traps of order 5 also necessitate that all bits in each group are treated together, because statistics of lower order are misleading.

3. *hTrap: Hierarchical Trap.* Dec-3 and trap-5 problems can be decomposed into separable subproblems of a fixed order. Nonetheless, hBOA can also solve problems that cannot be decomposed into subproblems of bounded order on a single level, but that must be solved hierarchically by building the solution from low-order building blocks over a number of levels of difficulty.

   Hierarchical traps (hTraps) [18] constitute a class hierarchical problems that cannot be efficiently solved using a single-level decomposition. Hierarchical traps are created by combining trap functions of order 3 over multiple levels of difficulty. For hTraps, the string length should be an integer power of 3, that is, $n = 3^l$. On the lowest level, groups of 3 bits contribute to the overall fitness using generalized 3-bit traps defined as follows:

   $$trap_3(u) = \begin{cases} f_{high} & \text{if } u = 3 \\ f_{low} - u\frac{f_{low}}{2} & \text{otherwise} \end{cases}, \qquad (3)$$

   where $f_{high} = 1$ and $f_{low} = 1 + 0.1/l$.

   Each group of 3 bits corresponding to one of the traps is then mapped to a single symbol on the next level; a 000 is mapped to a 0, a 111 is mapped to a 1, and everything else is mapped to the null symbol '-'. The bits on the next level again contribute to the overall fitness using 3-bit traps defined above, and the groups are mapped to an even higher level. This continues until the top level is evaluated that contains 3 bits total. However, on the top level, a trap with $f_{high} = 1$ and $f_{low} = 0.9$ is applied. Any group of bits containing the null symbol does not contribute to the overall fitness. To make the overall contribution of each level of the same magnitude, the contributions of traps on ith level from the bottom are multiplied by 3^i.

The task is to maximize the function. hTraps have many local optima, but only one global optimum in the string of all ones. Nonetheless, any single-level decomposition into subproblems of bounded order will lead away from the global optimum. That is why hTraps necessitate an optimizer that can build solutions hierarchically by juxtaposing good partial solutions over multiple levels of difficulty until the global optimum if found.

For more details on hierarchical traps and solving hierarchical optimization problems, please see [17]. Other hierarchical problems can be found in [32].

4.2 Experimental Methodology

To analyze scalability of SMB, for each problem, we performed experiments for a range of problem sizes ($n = 30$ to 210 with step 15 for dec-3 and trap-5; $n = 9$, 27, 81, and 243 for hTrap). For each problem and problem size, hBOA with SMB was tested for t_{sb} from 1 to 20 with step 1. For each problem, problem size, and value of t_{sb}, a minimum population size required to find the global optimum in 10 out of 10 independent runs was determined using the bisection method [18]. To reduce noise, for each setting, 10 bisection runs were performed. Therefore, for each problem, problem size, and value of t_{sb}, 100 successful independent runs were done. Each run was terminated either when the global optimum was found or when hBOA completed a large number of generations and it was unlikely that the algorithm would find the optimum in a reasonable time (the maximum number of iterations was estimated from preliminary experiments).

Binary tournament selection without replacement was used in all experiments and the window size in RTR was $w = \min\{n, N/20\}$ where n is the problem size and N is the population size. BDe metric with a penalty for complex networks as described in [18] was used in all experiments to measure quality of Bayesian networks with decision trees.

4.3 Results

The experimental results were processed to obtain the following quantities:

1. *Speedup of structure building for each problem, problem size, and structure-building period.* The speedup is averaged over the 100 independent runs for each particular setting. To compute the speedup of model building, all factors have been incorporated into the theoretical model for computational complexity of hBOA, including the effects of SMB on the population size and the number of iterations until convergence, which are both measured empirically. The resulting speedup thus represents the actual speedup of the structure-building procedure of hBOA.

2. *Optimal speedup of structure building for each problem and problem size.* For each problem and problem size, an optimal speedup is determined empirically by using the value of t_{sb} that maximizes the speedup of structure building.

3. *Evaluation slowdown for optimal speedup of structure building.* For each problem and problem size, the evaluation slowdown is determined for the best value of t_{sb}, which corresponds to the optimal speedup. The evaluation slowdown is defined as the factor by which the number of evaluations increases compared to the case with no SMB (that is, $t_{sb} = 1$).

4. *Scalability.* Scalability with various values of t_{sb} is evaluated to investigate the effects of SMB on the scalability of hBOA on decomposable problems. Specifically, scalability with $t_{sb} = 1$ and $t_{sb} = 10$ is evaluated and compared.

Figure 1 shows the structure-building speedup and the evaluation slowdown for dec-3, trap-5, and hTrap.

Figure 2 shows the optimal speedup of structure building and the corresponding evaluation slowdown for dec-3, trap-5, and hTrap.

Figure 3 shows the growth of the number of evaluations with problem size on dec-3, trap-5 and hTrap for $t_{sb} = 1$ (without SMB) and $t_{sb} = 10$ (with SMB where the structure is built in every 10th iteration).

4.4 Discussion of Results

The results presented in Figure 1 indicate that for all test problems, the structure-building speedup grows with the structure-building period t_{sb} until it reaches a local maximum at the optimal value of t_{sb} that maximizes the speedup; the rate of speedup growth increases with problem size. Additionally, these results indicate that the evaluation slowdown becomes less significant as the problem size increases and for larger problems, the evaluation slowdown remains nearly constant for all values of t_{sb}.

The results presented in Figure 2 indicate that the optimum speedup grows with problem size and its polynomial approximation grows as $\Theta(n^{0.26})$ to $\Theta(n^{0.65})$; that means that the maximum speedup of structure building is expected to grow with problem size and lead to a significant decrease of asymptotic complexity of model building in hBOA. On the other hand, the evaluation slowdown corresponding to the best value of t_{sb} is much less significant than the speedup of model building. In fact, the evaluation slowdown corresponding to the optimal speedup of model building decreases with problem size for both decomposable problems of bounded difficulty (dec-3 and trap-5), although it slowly increases for the hierarchical problem (hTrap).

The results presented in Figure 3 indicate that SMB does not negatively affect the polynomial growth of the number of evaluations until convergence with problem size; specifically, they show that with a fixed structure-building period $t_{sb} = 10$, the number of evaluations appears to grow with a polynomial of the same order as without SMB or even better.

Therefore, the results lead to four important observations:

1. SMB leads to a significant decrease of the asymptotic time complexity of model building given an appropriate period t_{sb} for rebuilding the structure.

2. The optimal speedup obtained with SMB grows with problem size.

3. The factor by which SMB increases the number of evaluations is insignificant compared to the speedup of model building and it decreases with problem size for decomposable problems of bounded difficulty.

4. SMB does not lead to an increase of the asymptotic complexity of hBOA with respect to the number of evaluations until convergence.

5. FUTURE WORK

One of the most important topics for future work is to design an adaptive schedule for SMB so that the frequency of structure building can be automatically controlled based on the problem at hand and the properties of the current population. SMB should also be analyzed on challenging real-world problems where the model building is indeed the bottleneck; the results presented in this paper indicate that SMB can be expected to be very effective. Theoretical and empirical models should be derived to provide bounds on the optimal value of the structure-building period for important classes of problems.

6. SUMMARY AND CONCLUSIONS

This paper presented and analyzed an efficiency enhancement technique called *sporadic model building*, which can be used to speed up model building in BOA, hBOA and other advanced EDAs that use complex multivariate probabilistic models. In sporadic model building, the structure of the probabilistic model is not updated in every iteration; instead, in some iterations only the parameters of the previous structure are updated with respect to the selected population of promising solutions. Building the model structure is the most computationally expensive part of hBOA variation and can become a computational bottleneck for large and complex problems. That is why sporadic model building represents an important efficiency enhancement technique for BOA, hBOA, and other advanced EDAs. The paper proposed a simple schedule for sporadic model building, where the updates of the model structure are done with a fixed period t_{sb} called the structure-building period. For example, $t_{sb} = 1$ denotes the scenario where the structure is updated in every iteration, whereas $t_{sb} = 3$ denotes the scenario where the structure is updated in every third iteration.

The experimental results presented in this paper lead to three important observations regarding the use of sporadic model building for solving nearly decomposable and hierarchical problems by hBOA:

1. Sporadic model building leads to significant improvements of asymptotic time complexity of hBOA model building and the optimal speedup of model building grows with problem size (here the optimal speedup can be approximated as $\Theta(n^{0.26})$ to $\Theta(n^{0.65})$ depending on the problem). The speedup of hBOA variation is upper bounded by $\Theta(n)$ because of parameter updates and sampling.

2. Sporadic model building leads to an increase in the number of evaluations until convergence but the factor by which sporadic model building increases the overall number of evaluations is insignificant and it decreases with problem size for decomposable problems of bounded difficulty.

3. Sporadic model building does not lead to an increase of the asymptotic complexity of hBOA with respect to the number of evaluations until convergence and the number of evaluations with a fixed value of t_{sb} remains a low-order polynomial for nearly decomposable and hierarchical problems.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0096, the National Science Foundation under NSF CAREER grant ECS-0547013, ITR grant DMR-03-25939 (at Materials Computation Center, UIUC), and ITR grant DMR-0121695 (at CPSD), the Dept. of Energy under grant DEFG02-91ER45439 (at Fredrick Seitz MRL), and the Research Award and the Research Board at the University of Missouri. Some experiments presented in this work were done using the hBOA software developed by Martin Pelikan and David E. Goldberg at the University of Illinois at Urbana-Champaign. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

7. REFERENCES

[1] L. A. Albert. Efficient genetic algorithms using discretization scheduling. Master's thesis, University of Illinois at Urbana-Champaign, Department of General Engineering, Urbana, IL, 2001.

[2] P. A. N. Bosman and D. Thierens. Linkage information processing in distribution estimation algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, I:60–67, 1999.

[3] D. M. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. Technical Report MSR-TR-97-07, Microsoft Research, Redmond, WA, 1997.

[4] K. Deb and D. E. Goldberg. Sufficient conditions for deceptive and easy binary functions. *Annals of Mathematics and Artificial Intelligence*, 10:385–408, 1994.

[5] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.

[6] D. E. Goldberg. *The design of innovation: Lessons from and for competent genetic algorithms*, volume 7 of *Genetic Algorithms and Evolutionary Computation*. Kluwer Academic Publishers, 2002.

[7] G. Harik. Linkage learning via probabilistic modeling in the ECGA. IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 1999.

[8] G. R. Harik. Finding multimodal solutions using restricted tournament selection. *Proceedings of the International Conference on Genetic Algorithms (ICGA-95)*, pages 24–31, 1995.

[9] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Research, Redmond, WA, 1994.

[10] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[11] R. A. Howard and J. E. Matheson. Influence diagrams. In R. A. Howard and J. E. Matheson, editors, *Readings on the principles and applications of decision analysis*, volume II, pages 721–762. Strategic Decisions Group, Menlo Park, CA, 1981.

[12] J. R. Koza. *Genetic programming: On the programming of computers by means of natural selection*. The MIT Press, Cambridge, MA, 1992.

[13] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer, Boston, MA, 2002.

[14] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature*, pages 178–187, 1996.

[15] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.

[16] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[17] M. Pelikan. *Bayesian optimization algorithm: From single level to hierarchy*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002. Also IlliGAL Report No. 2002023.

[18] M. Pelikan. *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms*. Springer-Verlag, 2005.

[19] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 511–518, 2001.

[20] M. Pelikan and D. E. Goldberg. A hierarchy machine: Learning to optimize from nature and humans. *Complexity*, 8(5):36–45, 2003.

[21] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, I:525–532, 1999.

[22] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[23] M. Pelikan, K. Sastry, and D. E. Goldberg. Scalability of the Bayesian optimization algorithm. *International Journal of Approximate Reasoning*, 31(3):221–258, 2002.

[24] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.

[25] K. Sastry. Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master's thesis, University of Illinois at Urbana-Champaign, Department of General Engineering, Urbana, IL, 2001. Also IlliGAL Report No. 2002004.

[26] K. Sastry and D. E. Goldberg. On extended compact genetic algorithm. IlliGAL Report No. 2000026, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 2000.

[27] K. Sastry, M. Pelikan, and D. E. Goldberg. Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation. *Proceedings of the IEEE Conference on Evolutionary Computation*, pages 720–727, 2004.

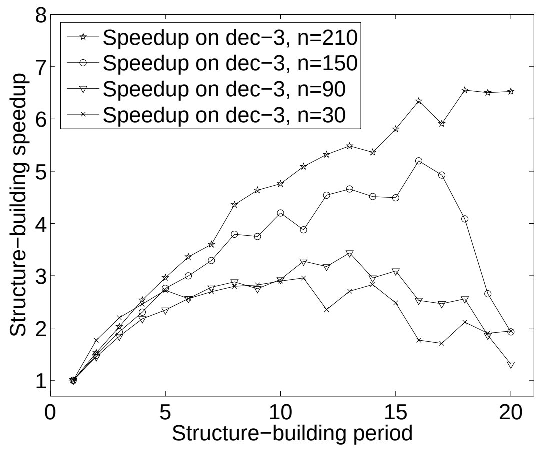
[28] A. Sinha and D. E. Goldberg. Verification and extension of the theory of global-local hybrids. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 591–597, 2001. Also IlliGAL Report No. 2001010.

[29] R. Srivastava and D. E. Goldberg. Verification of the theory of genetic and evolutionary continuation. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 551–558, 2001. Also IlliGAL Report No. 2001007.

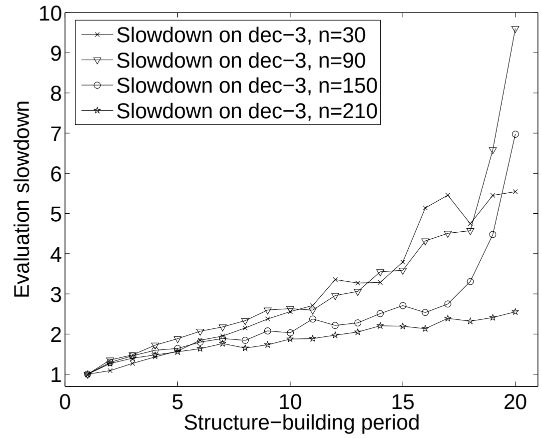
[30] D. Thierens. *Analysis and design of genetic algorithms*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1995.

[31] D. Thierens, D. E. Goldberg, and A. G. Pereira. Domino convergence, drift, and the temporal-salience structure of problems. *Proceedings of the International Conference on Evolutionary Computation (ICEC-98)*, pages 535–540, 1998.

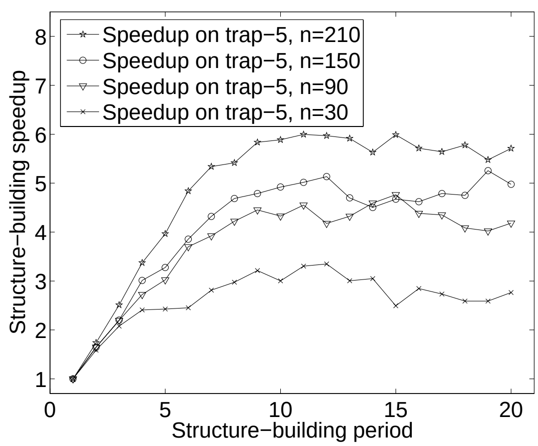
[32] R. A. Watson, G. S. Hornby, and J. B. Pollack. Modeling building-block interdependency. *Parallel Problem Solving from Nature*, pages 97–106, 1998.



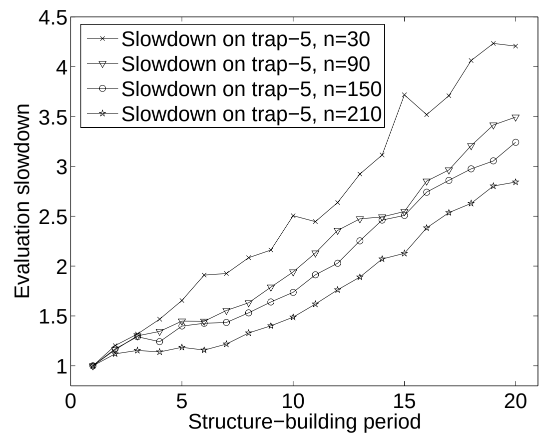
(a) Speedup of structure building for dec-3.



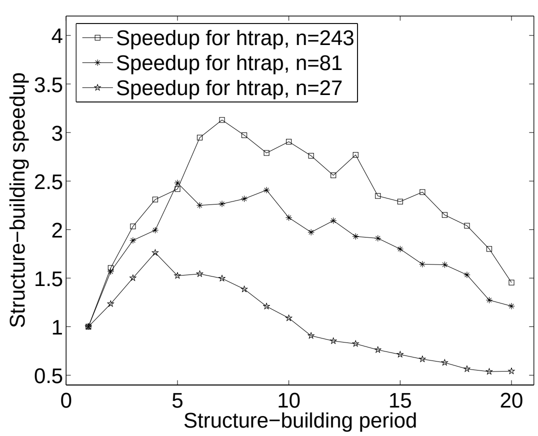
(b) Slowdown of evaluation for dec-3.



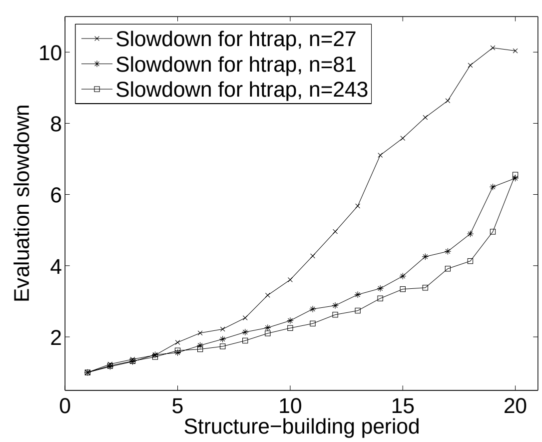
(c) Speedup of structure building for trap-5.



(d) Slowdown of evaluation for trap-5.

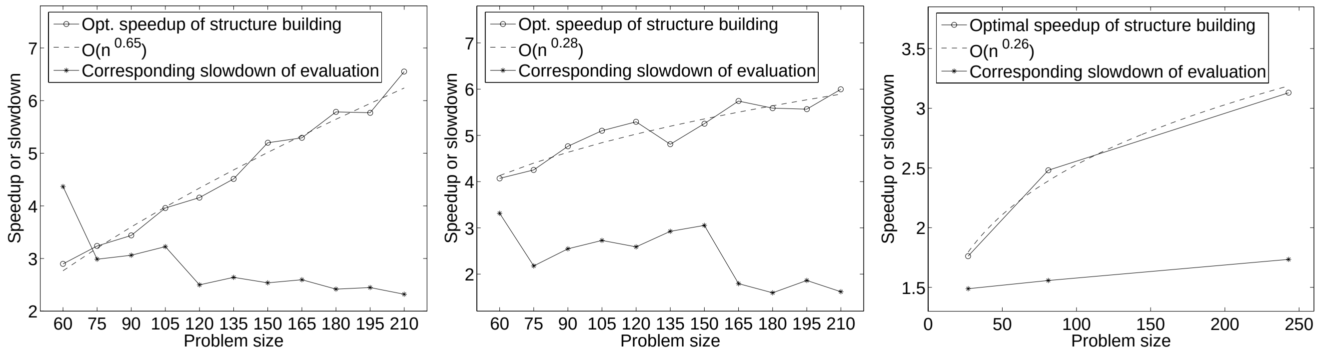


(e) Speedup of structure building for hTrap.



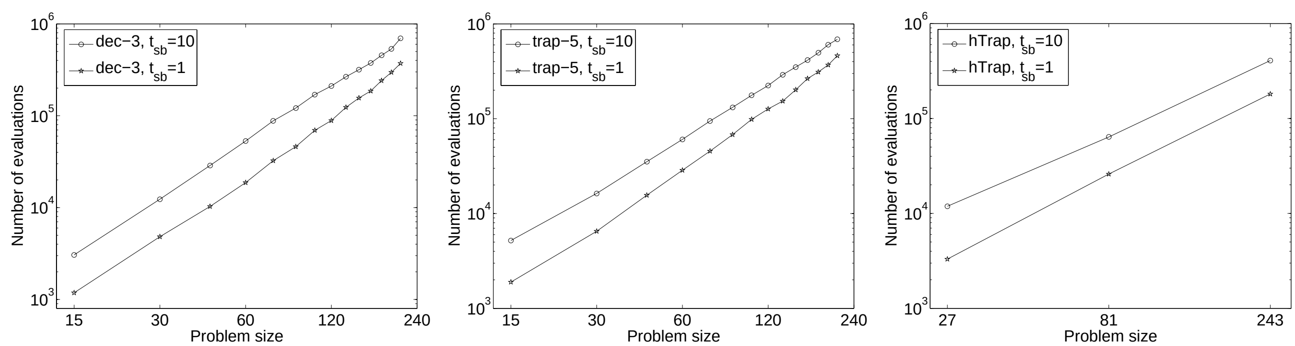
(f) Slowdown of evaluation for hTrap.

Figure 1: The speedup of structure building and the slowdown of evaluation for dec-3, trap-5 and hTrap.



(a) Dec-3 (b) Trap-5 (c) hTrap

Figure 2: The optimal speedup of structure building and the corresponding evaluation slowdown for dec-3, trap-5, and hTrap.



(a) Dec-3 (b) Trap-5 (c) hTrap

Figure 3: The growth of the number of evaluations for dec-3, trap-5 and hTrap with $t_{sb} = 1$ (build structure always) and $t_{sb} = 10$ (build structure in every 10th iteration).