2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

Diversity Allocation for Dynamic Optimization using the Extended Compact Genetic Algorithm

Chung-Yao Chuang
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA
Email: cychuang@cmu.edu

Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA
Email: sfs@cs.cmu.edu

***Abstract*—This paper investigates the issues of maintaining diversity in the Extended Compact Genetic Algorithm (ECGA) for handling Dynamic Optimization Problems (DOPs). Specifically, we focused on how a diversity maintenance mechanism places samples in the search space, and derive an approach that is more appropriate for DOPs that change progressively. The discussion proceeds in two parts. First, we reaffirm the perspective that the problem structure should be considered when maintaining diversity for addressing DOPs. This point is demonstrated by an additively decomposable DOP in which each subfunction has two complementary optima. Following that, we further discuss how we can better allocate the samples for DOPs that change progressively by thinking about the current promising region, which should contain the current optima, and its neighborhood. Based on this notion, we devise a mechanism that utilizes the information provided by the probabilistic models from ECGA and uses a trade-off between exploration and exploitation to achieve the desired diversity allocation. The empirical results show that our approach follows the changing optima better compared to techniques that use Restricted Tournament Replacement (RTR). Furthermore, it requires only half of the function evaluations needed by approaches that use RTR.**

I. Introduction

Estimation of Distribution Algorithms (EDAs) [1], [2], [3] are a class of evolutionary algorithms that replace the traditional variation operators, such as mutation and crossover, by the procedure of building a probabilistic model on promising solutions and sampling the constructed model to generate new candidate solutions. By using probabilistic models to summarize the information, advanced EDAs are able to incorporate techniques from machine learning and statistics to automatically discover the multivariate interactions between problem variables, which leads to an approximation of problem decomposition and the recognition of important substructures that constitute the promising solutions.

In the past studies, EDAs have demonstrated their usefulness in solving problems that were intractable with other evolutionary algorithms or achieving significantly better results compared to other techniques [4]. However, their application to Dynamic Optimization Problems (DOPs) has been rather limited [5], [6], [7], [8], [9], [10], [11], [12], [13]. Moreover, most of the research has been focused on univariate EDAs [6], [9], [10], [11], [12], which assume that problem variables are independent of each other and comprise no structure learning capability. In this paper, we focus on using a multivariate EDA, called Extended Compact Genetic Algorithm (ECGA) [14], to address the DOPs.

There is some previous research on using ECGA for DOPs. To handle DOPs properly, it is important to maintain or introduce diversity so that the population can respond to the changing environment quickly. In [5], ECGA was first modified to handle problems with non-stationary fitness landscape. This approach is based on reinitializing the population after each change so that the diversity can be increased at the beginning of the new environment. This research demonstrated a way to use learned structural information about the problem to accelerate the growth of highly-fit substructures so that it can adapt to the new environment more promptly. This work was later extended [7] to include substructural niching [8]. In substructural niching, niches are defined within the linkage group rather than at the individual level. After the corresponding schema average fitness [15] is calculated for each substructure, the sampling probabilities are changed based on their associated fitness. While this methodology can be used to maintain diversity in the population, it is used in [8] as a way to speed up the propagation of better substructures, using the reinitialization of population as the sole source to provide diversity.

Moreover, the approaches proposed in [5], [7], [8] assume the signal to a change is given, so that it can introduce diversity by restarting the population. In this work, we only discuss approaches that maintain diversity throughout the run so that ECGA can respond to changes without requiring additional means to detect alteration in the fitness landscape. Two approaches within this category were proposed in [13], which are based on Restricted Tournament Replacement (RTR). In the first part of this paper, we will use these two RTR-based approaches to demonstrate the idea that the problem structure should be considered when maintaining diversity for DOPs. Different from previous research that also advocates this perspective (e.g. [13]), we propose a more illustrative test problem that shows clearly the necessity of such consideration.

After reaffirming the utility of using structural information about the problem, we will further discuss how we can allocate our samples in the search space to better address DOPs that change progressively. This means that we are considering the

DOPs in which the optima, although moving in unpredictable direction, will change to some neighboring point in the search space in the next changing cycle. This assumption corresponds to a class of problems in which its fitness landscape changes gradually and frequently, and keeping pace with these partial changes is key to the performance of the optimizer. It should be noted that if the environment changes unboundedly or drastically, on average no method will outperform restarting the optimizer from scratch every time a change occurs.

The rest of this paper is organized as follows. The next section gives a brief review of ECGA. After that, Section III provides a description of two RTR-based approaches proposed in [13]. In Section IV, an additive decomposable DOP is proposed to demonstrate that the problem structure should be considered when maintaining diversity for handling DOPs. Section V discusses how we can allocate the samples in the search space to better handle DOPs that changes progressively, and proposes a mechanism to achieve the desired diversity allocation. Finally, Section VI concludes this paper.

II. Extended Compact Genetic Algorithm

The extended compact genetic algorithm (ECGA) [14] uses a product of marginal distributions on a partition of the variables. This kind of probability distribution belongs to a class of probabilistic models known as marginal product models (MPMs). In this kind of model, subsets of variables can be modeled jointly, and each subset is considered independent of other subsets. In this work, the conventional notation is adopted that variable subsets are enclosed in brackets. Table I presents an example of MPM defined over four variables: X_1, X_2, X_3 and X_4. In this example, X_2 and X_4 are modeled jointly and each of the three variable subsets ($[X_1]$, $[X_2\ X_4]$ and $[X_3]$) is considered independent of other subsets. For instance, the probability that this MPM generates a sample $X_1X_2X_3X_4 = 0101$ is calculated as follows,

$$\begin{aligned} & P(X_1X_2X_3X_4 = 0101) \\ & = P(X_1 = 0) \times P(X_2 = 1, X_4 = 1) \times P(X_3 = 0) \\ & = 0.4 \times 0.4 \times 0.5. \end{aligned}$$

In fact, as its name suggests, a marginal product model represents a distribution that is a “product” over the marginal distributions defined over variable subsets.

In the ECGA, both the structure and the parameters of the model are searched and optimized with a greedy approach to fit the statistics of the selected set of promising solutions. The measure of a good MPM is quantified based on the minimum description length (MDL) principle [16], which assumes that given all things are equal, simpler distributions are better than complex ones. The MDL principle thus penalizes both inaccurate and complex models, thereby, leading to a near-optimal distribution. Specifically, the search measure is the MPM complexity which is quantified as the sum of model complexity, C_m, and compressed population complexity, C_p. The greedy MPM search first considers all variables as independent and each of them forms a separate variable subset. In each iteration, the greedy search merges two variable subsets that yields the most $C_m + C_p$ reduction. The process continues until there is no further merge that can decrease the combined complexity.

The model complexity, C_m, quantifies the model representation in terms of the number of bits required to store all the marginal distributions. Suppose that the given problem is of length ℓ with binary encoding, and the variables are partitioned into m subsets with each of size k_i, $i = 1 \dots m$, such that $\ell = \sum_{i=1}^{m} k_i$. Then the marginal distribution corresponding to the ith variable subset requires $2^{k_i} - 1$ frequency counts to be completely specified. Taking into account that each frequency count is of length $\log_2(n+1)$ bits, where n is the population size, the model complexity, C_m, can be defined as

$$C_m = \log_2(n+1) \sum_{i=1}^{m} \left(2^{k_i} - 1\right) \ .$$

The compressed population complexity, C_p, quantifies the suitability of the model in terms of the number of bits required to store the entire selected population (the set of promising solutions picked by selection operator) with an ideal compression scheme applied. The compression scheme is based on the partition of the variables. Each subset of the variables specifies an independent “compression block” on which the corresponding partial solutions are optimally compressed. Theoretically, the optimal compression method encodes a message of probability p_i using $-\log_2 p_i$ bits. Thus, taking into account all possible messages, the expected length of a compressed message is $\sum_i -p_i \log_2 p_i$ bits, which is optimal. In information theory [17], the quantity $-\log_2 p_i$ is called the *information* of that message and $\sum_i -p_i \log_2 p_i$ is called the *entropy* of the corresponding distribution. Based on information theory, the compressed population complexity, C_p, can be derived as

$$C_p = n \sum_{i=1}^{m} \sum_{j=1}^{2^{k_i}} -p_{ij} \log_2 p_{ij} \ ,$$

where p_{ij} is the frequency of the jth possible partial solution to the ith variable subset observed in selected population.

III. Restricted Tournament Replacement

The Restricted Tournament Replacement (RTR) [18] is a niching method that has been used successfully as a diversity maintenance mechanism in EDAs [19]. In RTR, each newly generated solution $\mathbf{x}$ will be incorporated into the current population by the following procedure:

1) Randomly select a set of solutions Y of size w from the current population.
2) Find the solution $\mathbf{y}$ in Y that is most similar to $\mathbf{x}$ in terms of Hamming distance.
3) Replace $\mathbf{y}$ with $\mathbf{x}$ if $\mathbf{x}$ is better, otherwise discard $\mathbf{x}$.

The window size w is usually set to the problem size, as suggested in [19]. RTR is incorporated into ECGA [13] as follows: after sampling the MPM to generate new solutions, those newly created solutions are incorporated one by one

TABLE I

AN EXAMPLE OF MARGINAL PRODUCT MODEL THAT DEFINES A JOINT DISTRIBUTION OVER FOUR VARIABLES. THE VARIABLES ENCLOSED IN THE SAME BRACKETS ARE CONSIDERED DEPENDENT AND MODELED JOINTLY. EACH VARIABLE SUBSET IS CONSIDERED INDEPENDENT OF OTHER SUBSETS.

$[X_1]$	$[X_2\ X_4]$	$[X_3]$
$P(X_1=0)=0.4$	$P(X_2=0,X_4=0)=0.4$	$P(X_3=0)=0.5$
$P(X_1=1)=0.6$	$P(X_2=0,X_4=1)=0.1$	$P(X_3=1)=0.5$
	$P(X_2=1,X_4=0)=0.1$	
	$P(X_2=1,X_4=1)=0.4$	

into the current population using RTR. Note that in dealing with DOPs, the fitness of the original population has to be reevaluated before RTR takes place, so that the fitness values are properly updated in case that a change occurs. Thus the consumption of function evaluations is twice the amount of that is used when solving stationary problems.

It was further proposed in [13] to use, instead of Hamming distance, a similarity metric based on substructural distance. The substructural distance between two solutions is defined as the number of substructures in which they differ, according to the problem decomposition provided by the MPM. For example, suppose the MPM built by ECGA has structure $[X_1X_2X_3X_4][X_5X_6X_7X_8][X_9X_{10}X_{11}X_{12}]$, assuming a 12-bit problem. When comparing a newly created solution:

$\mathbf{x}$: `1111-0000-1111`

with two solutions in the current population:

$\mathbf{y}_1$: `0000-0000-1111`, and

$\mathbf{y}_2$: `0111-0100-0111`,

the measure based on substructure assigns the distance between $\mathbf{x}$ and $\mathbf{y}_1$ to be 1 (they only differ in the first substructure), and the distance between $\mathbf{x}$ and $\mathbf{y}_2$ to be 3. In comparison, the Hamming distance between $\mathbf{x}$ and $\mathbf{y}_1$ is 4, and the Hamming distance between $\mathbf{x}$ and $\mathbf{y}_2$ is 3.

This illustrates that considering similarity at the substructural level is very different from using Hamming distance, and it will have a better effect in preserving substructure-wise diversity. In the next section, we will show that maintaining diversity in a substructure-wise fashion is important for addressing DOPs. Different from previous works that also possess such belief, we will provide a more evident demonstration by proposing an additively decomposable DOP that overwhelms RTR using Hamming distance.

IV. DUAL-PEAK PROBLEM

This section serves to reaffirm the perspective that maintaining diversity at a substructural level is beneficial, sometimes even necessary, for addressing DOPs. This idea is demonstrated by comparing the performance of two different kinds of RTRs described in the previous section. We use an additively decomposable DOP, in which each subfunction has two complementary optima, to show that overlooking the problem structure when maintaining diversity could be unfavorable to the performance.

Specifically, we use a k-bit *dual-peak function*, which can be specified by its two complementary optima $s_1s_2\cdots s_k$ and $\bar{s_1}\bar{s_2}\cdots\bar{s_k}$, to construct our test problem. A dual-peak function is defined as

$$d_{s_1s_2...s_k}(x_1x_2...x_k) = \begin{cases} k, & \text{if } x_i = s_i \text{ for all } i = 1...k \\ k, & \text{if } x_i = \bar{s_i} \text{ for all } i = 1...k \\ 1, & \text{otherwise} \end{cases}$$

where $\bar{s_i}$ is the inverse of s_i ($\bar{0} = 1$ and $\bar{1} = 0$). For example, $d_{1100}(\mathbf{x})$ has two optima at 1100 and 0011. By concatenating ten 4-bit dual-peak functions, we construct a 40-bit test problems,

$$f(x_1x_2\cdots x_{40}) = d_{s_1\cdots s_4}(x_1\cdots x_4) + d_{s_5\cdots s_8}(x_5\cdots x_8) + \cdots + d_{s_{37}\cdots s_{40}}(x_{37}\cdots x_{40})$$

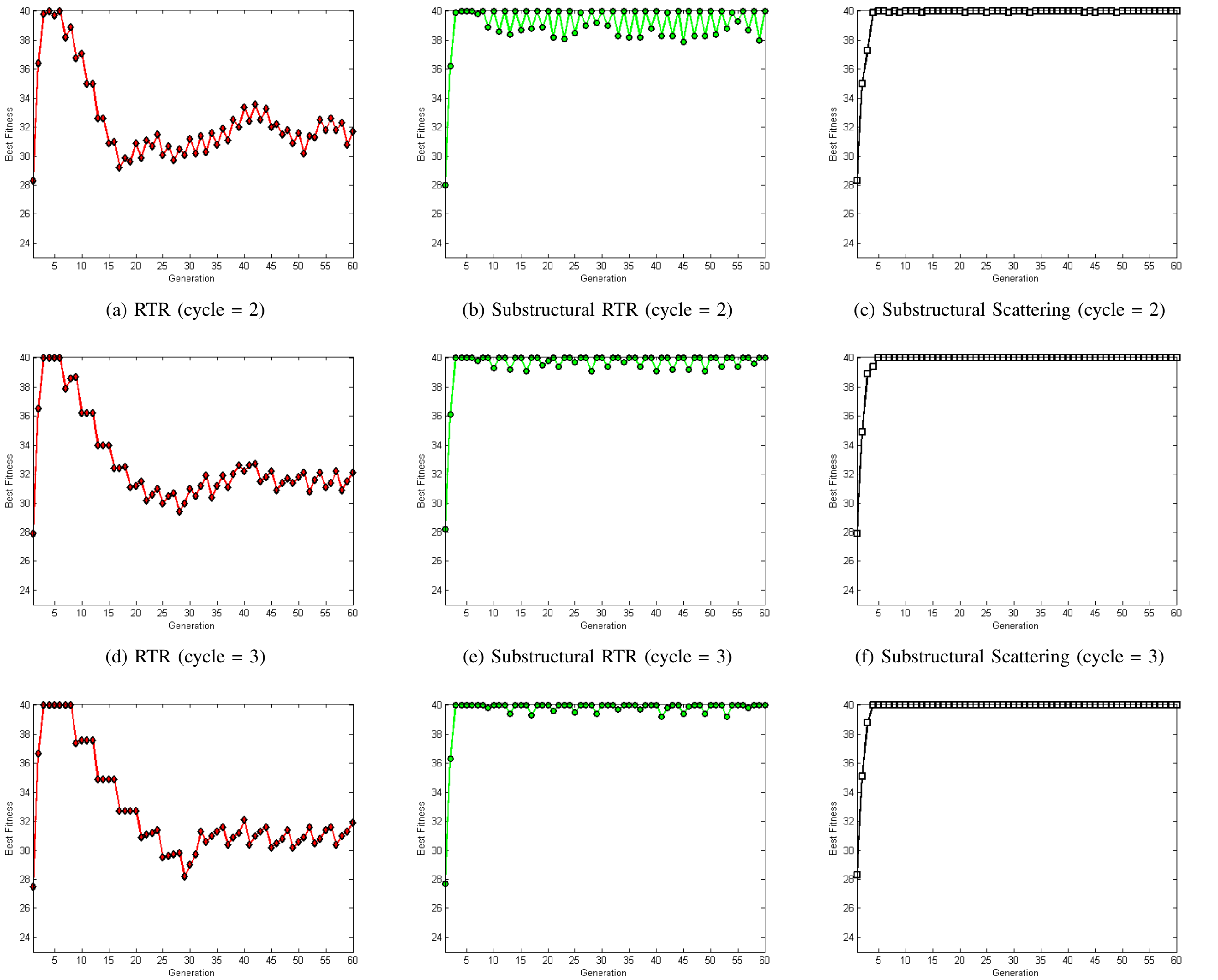
where $s_1s_2\cdots s_{40}$ are set to all zeros initially. To make the problem dynamic, at each environmental shift, one subfunction is randomly picked and its optima are changed to some other values randomly (i.e., the corresponding s_i's are set to some other values.) Note that the design of this test problem is in accord with our notion of DOP that changes progressively as described in Section I.

Using the DOP defined above, we test the ECGA with two different kinds of RTRs described in the previous section. We want to observe the ability of each method to track the changing global optima. To simulate different pace of variation, we use changing cycles range from 2 to 4 generations. For all experiments, the population size is set to $n = 4000$, and we use tournament selection with tournament size 16. Each run terminates after 60 time steps (generations). All the results are averaged over 30 independent runs.

The empirical results are presented in Table II and Figure 1*. It can be observed from the figures that after 2 or 3 changing cycles, ECGA with RTR lost track of the global optima, and cannot recover from that. On the other hand, ECGA with substructural RTR gave much better performance. Quantitatively, Table II gives a comparison between them in terms of the number of generations that each approach obtained the global optima. It can be seen that there is a huge difference between maintaining diversity in a substructure-wise fashion and using Hamming distance in dealing with this problem.

Clearly the dual-peak problem overwhelms the ability of RTR in maintaining diversity. The cause of this can be explained with the example presented in the previous section. If we use Hamming distance, $\mathbf{x}$ will replace $\mathbf{y}_2$ instead of

*The third approach, substructural scattering, will be described in Section V. Here we discuss the difference between RTR and substructural RTR, and its implication that the problem structure should be considered when maintaining diversity for addressing DOPs.



(a) RTR (cycle = 2) (b) Substructural RTR (cycle = 2) (c) Substructural Scattering (cycle = 2)

(d) RTR (cycle = 3) (e) Substructural RTR (cycle = 3) (f) Substructural Scattering (cycle = 3)

(g) RTR (cycle = 4) (h) Substructural RTR (cycle = 4) (i) Substructural Scattering (cycle = 4)

Fig. 1. Empirical results for the dual-peak problem. Three different changing cycles (from 2 to 4) are experimented to simulate different paces of variation. The figures shows the results averaged over 30 runs.

replacing $\mathbf{y}_1$ (assuming $\mathbf{x}$ is better), thus losing the diversity in a substructure-wise sense. Moreover, two optima to the dual-peak function are complementary, so a distance of k is added every time we compare two different optimal substructures using Hamming distance, which leads to the situation that this kind of destructive replacement happens more frequently. This inaccuracy in replacement is accumulated and eventually the RTR will be unable to preserve the substructures needed to deal with the change.

This experiment demonstrates an instance in which it is necessary to consider problem structure when maintaining diversity. Note that the stationary version of the dual-peak problem can be easily solved by ECGA, so the source of difficulty resides in the dynamic property of the problem. It demonstrates a way to mislead the diversity maintenance mechanism that overlooks the structural information of the problem, and shows that without such consideration, the operation can be destructive.

TABLE II
THE NUMBER OF GENERATIONS IN WHICH EACH APPROACH OBTAINED THE GLOBAL OPTIMA FOR THE DUAL-PEAK PROBLEM. A RUN LASTS FOR 60 GENERATIONS. THE RESULTS ARE AVERAGED OVER 30 RUNS.

change cycle (generations)	RTR	Substructural RTR	Substructural Scattering
2	6.2/60	45.2/60	56.7/60
3	7.1/60	54.2/60	57.4/60
4	7.6/60	55.7/60	57.6/60

V. Diversity Allocation

In the previous section, we established that the problem structure should be considered when maintaining diversity in the context of DOPs, i.e., diversity should be maintained at a substructural level. In this section, we further discuss how we can allocate our samples in the search space to better address DOPs that change progressively. Similar to the previous section, this discussion will be demonstrated by a test problem. This test problem is made intentionally simple to better reflect our perspective.

Specifically, we use a k-bit *single-peak function*, which has only one optimum and is flat elsewhere, to construct our test problem. A single-peak function is defined as

$$h_{s_1 s_2 \ldots s_k}(x_1 x_2 \ldots x_k) = \begin{cases} k, & \text{if } x_i = s_i \text{ for all } i = 1 \ldots k \\ 1, & \text{otherwise} \end{cases}$$

where $s_1 s_2 \cdots s_k$ is the optimum to the function. Again, we use ten 4-bit single-peak functions as sunfunctions to construct our test problem,

$$g(x_1 x_2 \cdots x_{40}) = h_{s_1 \cdots s_4}(x_1 \cdots x_4) + h_{s_5 \cdots s_8}(x_5 \cdots x_8) + \cdots + h_{s_{37} \cdots s_{40}}(x_{37} \cdots x_{40})$$

where $s_1 s_2 \cdots s_{40}$ are set to all zeros initially. As before, at each changing cycle, one subfunction will be selected at random and its optimum will be changed randomly to some other value. Compared to dual-peak problem, this problem is simpler in the sense that at any given time step, there is only one global optimum in the search space. Thus, it allows us to discuss the promising region and its neighborhood more intuitively.

Our first step of analysis is based on the observation of how substructural RTR allocates samples in the search space. Figure 2 shows that after 30 generations, the distribution of the solutions according to their fitness values when solving the single-peak problem. It can be seen that a large portion of the population is of low fitness values. This is because the aim of substructural RTR is to keep as many different substructures as possible in the population, so the low fitness portion corresponds to the solutions that are composed of mostly non-optimal substructures. However, if our goal is to address DOPs that change progressively, such diversity preservation may be excessive. Recall that our assumption for this type of DOPs is that the optimum, although moving in some unpredictable direction, will most likely change to some neighboring point in the search space in the next environmental shift. To account for this kind of gradual change, it is more beneficial to place samples around the current promising region than spending our search effort on low fitness area, because in this way, we are more likely to keep pace with the changing optimum.

This perspective fits nicely into the framework of EDA, because the probabilistic model built by EDA can be seen as a description of our current belief of where the promising region is. To produce the desired allocation of samples, i.e., dense around the current promising region and sparse in the distant area, we can generate samples using a distribution that is slightly diverted from the one encoded in the probabilistic model to cover the vicinity of the current promising region. In ECGA, we can achieve this in a principled way: assuming the MPM is composed of m marginal models (i.e., a solution is decomposed into m substructures), we can devise an "exploration" mechanism that creates some of the substructures in a solution *not* by sampling the constructed MPM, but by generating them uniformly randomly.

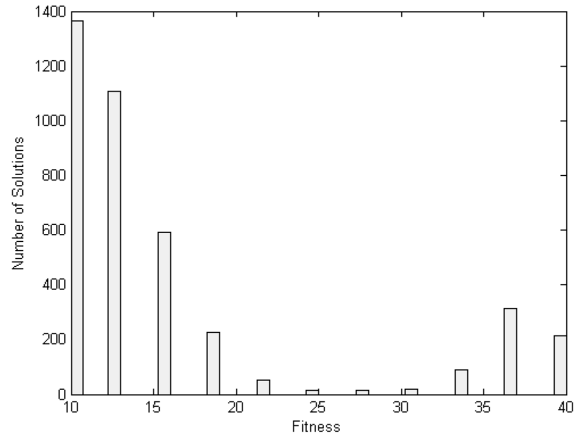


Fig. 2. Fitness distribution of substructural RTR when dealing with the single-peak problem.

Of course, any good search algorithm will try to balance between exploration and exploitation. In ECGA, exploitation can be defined as sampling the constructed MPM, which corresponds to placing samples in the current promising region. To incorporate our mechanism of exploration, we introduce a tunable trade-off between exploration and exploitation. Suppose an "exploitation rate" ξ $(0 \leq \xi \leq 1)$ is specified, then it could be saying that in the next generation, approximately ξ of the population should be created by sampling the MPM. Now let the probability of invoking the exploration mechanism be p, we have that

$$(1-p)^m = \xi.$$

which says that the probability that a solution is created entirely according to the built model is ξ. Base on that, we can derive

$$p = 1 - \xi^{1/m}.$$

In implementation, when composing a new solution, most of the time we generate each substructure by sampling the corresponding marginal model in the MPM, but with probability p as derived above (based on specified ξ), we place a partial solution that is generated uniformly randomly for that substructure. In this way, we can control the trade-off between exploration and exploitation (by specifying ξ), and make the "exploration points" dense near the current promising region but sparse in more distant area (on average, only mp substructures in a solution are not generated according to the constructed MPM.) We call this approach *substructural scattering* because our goal is to distribute the search effort to cover the vicinity of the current promising region in a manner where the samples are like being dispersed from the promising region.

Figure 3 gives a qualitative examination of our approach. It shows that after 30 generations, the distribution of the solutions according to their fitness values when solving the single-

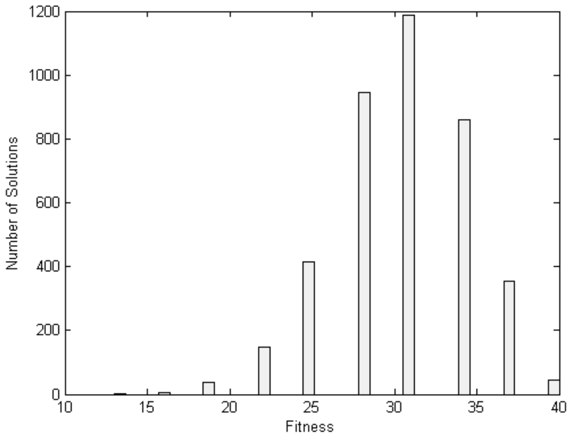


Fig. 3. Fitness distribution of substructural scattering when dealing with the single-peak problem. The exploitation rate is set to $\xi = 1/16$.

TABLE III
THE NUMBER OF GENERATIONS IN WHICH EACH APPROACH OBTAINED THE GLOBAL OPTIMA FOR THE SINGLE-PEAK PROBLEM. A RUN LASTS FOR 60 GENERATIONS. THE RESULTS ARE AVERAGED OVER 30 RUNS.

change cycle (generations)	RTR	Substructural RTR	Substructural Scattering
2	36.0/60	42.9/60	46.2/60
3	47.0/60	49.1/60	54.0/60
4	49.8/60	53.3/60	56.0/60

peak problem. Different from that when using substructural RTR, now the solutions are mostly allocated at one to four substructures away from the global optimum.

To empirically verify the effectiveness of our method in following the changing optimum, we compared the performance of it with the RTR-based approaches. The experimental settings are the same as what we used for the dual-peak problem. For substructural scattering, the exploitation rate is set to $\xi = 1/16$ †. The results are presented in Table III and Figure 4. It can be seen from Table III that substructural scattering performed better compared to the other two RTR-based methods. It obtained the global optimum more frequently under all three different paces of change. Moreover, Figure 4 shows that our approach is able to recover more consistently than the other two methods. To be more comprehensive about the performance of substructure scattering, we also include the experiment results of it when solving the dual-peak problem in the third column of Table II and Figure 1(c), (f) and (i).

Furthermore, because substructural scattering does not use any replacement operation as the two RTR-based approaches do (in each generation, substructural scattering generates a population anew), there is no extra cost on reevaluating previous population in order to incorporate new solutions. Thus, the consumption of fitness evaluation for substructural scattering is half of that required by RTR or substructural RTR.

†The rationale behind this choice is that we use tournament selection with tournament size 16. Therefore the MPM is built roughly based on 1/16 of the current population. Thus, it might be a good idea to place 1/16 of the next population in the current promising region. Although we don't have any theoretical justification for this yet, it shows good performance empirically.

VI. CONCLUSION

In this paper, we looked into two issues regarding the diversity maintenance for addressing dynamic optimization problems. First, we discussed the perspective that the problem structure should be taken into account when maintaining diversity, and used an additively decomposable DOP called dual-peak problem to demonstrated this situation. Following that, we continued to discuss how we can better allocate the samples in the search space to deal with DOPs that change progressively. This idea is based on considering the current promising region, and how we can place samples in its vicinity to better keep pace with the changing optimum. We designed a mechanism with a tunable trade-off between exploration and exploitation to achieve the desired diversity allocation. The empirical results showed that this method outperformed the two RTR-based approaches in terms of the ability to follow the changing optimum.

As a future work, we are interested in how to automatically adjust the trade-off between exploration and exploitation. Specifically, we would like to investigate whether we can use the change in the fitness of the population, along with the information provided by the probabilistic model, to control the ratio of the exploration operation. Hopefully, this would lead to a mechanism that gives a more reliable and responsive performance.

ACKNOWLEDGMENT

The work reported in this paper was funded in part by the Heinz Endowments under grant #1030675 and the CMU Robotics Institute.

REFERENCES

[1] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. binary parameters," in *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 1996, pp. 178–187.

[2] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, ser. Genetic algorithms and evolutionary computation. Boston, MA: Kluwer Academic Publishers, October 2001, vol. 2, ISBN: 0-7923-7466-5.

[3] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," *Computational Optimization and Applications*, vol. 21, no. 1, pp. 5–20, 2002.

[4] M. Pelikan, M. Hauschild, and F. Lobo, "Introduction to estimation of distribution algorithms," Missouri Estimation of Distribution Algorithms Laboratory (MEDAL), MEDAL Report 2012003, 2012.

[5] H. Abbass, K. Sastry, and D. Goldberg, "Oiling the wheels of change: The role of adaptive automatic problem decomposition in non–stationary environments," University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, IlliGAL Report 2004029, 2004.

[6] A. Ghosh and H. Muehlenbein, "Univariate marginal distribution algorithms for non-stationary optimization problems," *Int. J. Know.-Based Intell. Eng. Syst*, vol. 8, no. 3, pp. 129–138, 2004.

[7] K. Sastry, H. Abbass, and D. Goldberg, "Sub-structural niching in non-stationary environments," *AI 2004: Advances in Artificial Intelligence*, pp. 241–244, 2004.

[8] K. Sastry, H. Abbass, D. Goldberg, and D. Johnson, "Sub-structural niching in estimation of distribution algorithms," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 2005, pp. 671–678.

[9] S. Yang and X. Yao, "Experimental study on population-based incremental learning algorithms for dynamic optimization problems," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 9, no. 11, pp. 815–834, 2005.

[10] S. Yang, "Memory-enhanced univariate marginal distribution algorithms for dynamic optimization problems," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 3. IEEE, 2005, pp. 2560–2567.

[11] ——, "Population-based incremental learning with memory scheme for changing environments," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 2005, pp. 711–718.

[12] C. Fernandes, C. Lima, and A. Rosa, "UMDAs for dynamic optimization problems," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 399–406.

[13] C. Lima, C. Fernandes, and F. Lobo, "Investigating restricted tournament replacement in ECGA for non-stationary environments," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 439–446.

[14] G. Harik, "Linkage learning via probabilistic modeling in the ECGA," University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, IlliGAL Report 99010, 1999.

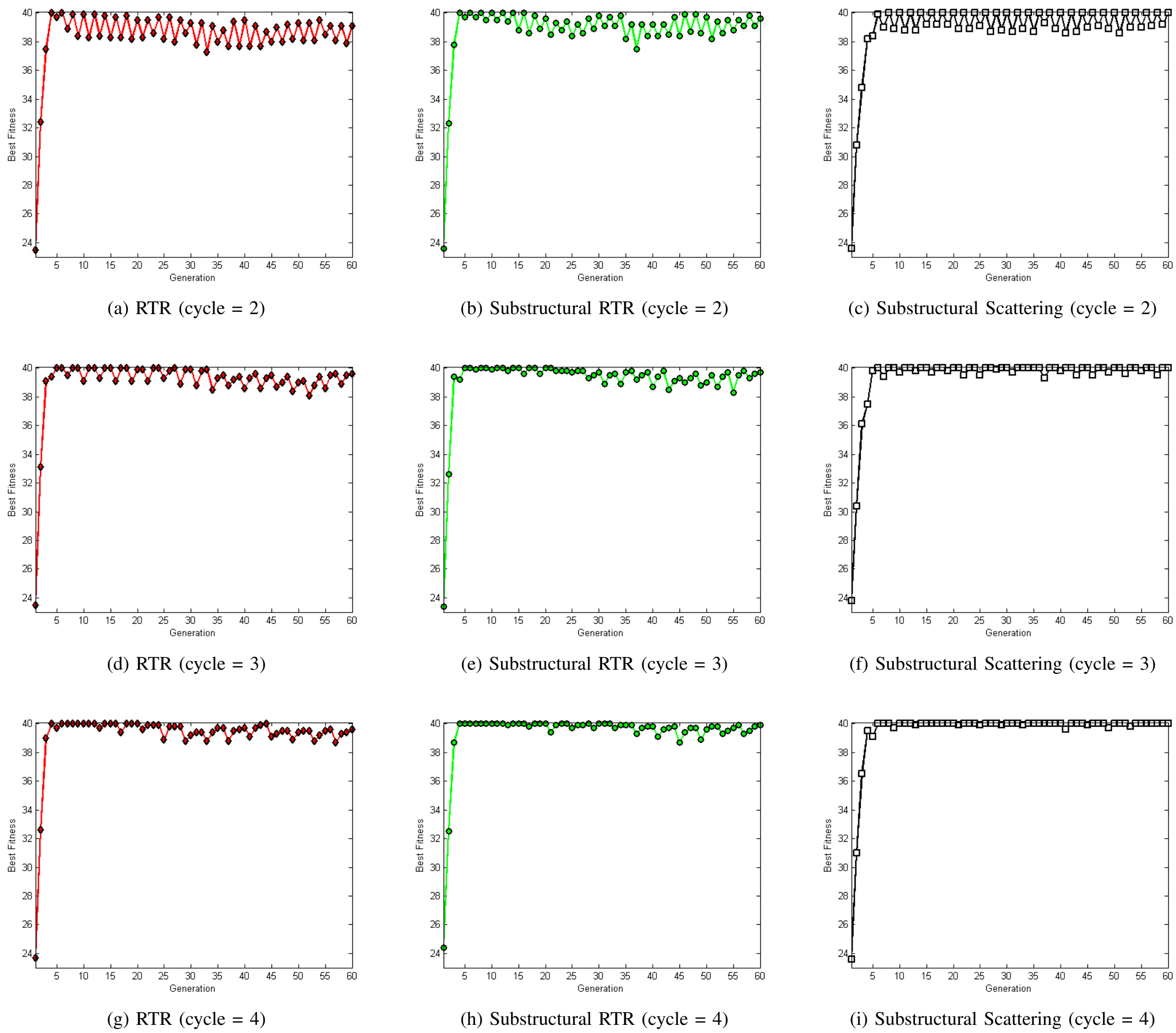
[15] K. Sastry, M. Pelikan, and D. Goldberg, "Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 1. IEEE, 2004, pp. 720–727.

[16] J. Rissanen, "Modelling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[17] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.

[18] G. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proceedings of the Sixth International Conference on Genetic Algorithms*. San Francisco, CA, 1995, pp. 24–31.

[19] M. Pelikan, *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*. Springer, 2005, vol. 170.



(a) RTR (cycle = 2) (b) Substructural RTR (cycle = 2) (c) Substructural Scattering (cycle = 2)

(d) RTR (cycle = 3) (e) Substructural RTR (cycle = 3) (f) Substructural Scattering (cycle = 3)

(g) RTR (cycle = 4) (h) Substructural RTR (cycle = 4) (i) Substructural Scattering (cycle = 4)

Fig. 4. Empirical results for the single-peak problem. Three different changing cycles (from 2 to 4) are experimented to simulate different paces of variation. The figures shows the results averaged over 30 runs.