In D. Keymeulen, A. Stoica, J. Lohn, and R. S. Zebulum, editors, *The Third NASA/DoD Workshop on Evolvable Hardware—EH-2001*, Long Beach, 12–14 July 2001. IEEE Computer Society, Los Alamitos, pp. 14–20, 2001.

Scouting Context-sensitive Components

Jeffrey O. Pfaffmann and Klaus-Peter Zauner

Department of Computer Science
Wayne State University
Detroit, Michigan, 48201, U.S.A.
biocomputing@cs.wayne.edu
fax.: +313-577-6868, ph.: +313-577-2477

Abstract

Nature's gadgets are implemented without being planned and therefore can utilize context-sensitive components. Thus functionality that would require extensive networks of context-free components can be elicited from a minimum of material. Proteins can serve as context-sensitive components for pattern processing applications. We here describe an evolutionary search strategy currently under investigation for its potential use in conjunction with computer controlled fluidics to evaluate the computational capabilities of proteins. Our algorithm employs evolutionary search not to seek an optimum, but to seek surprises. It directs experiments and incrementally constructs an empirical model from their outcome. Reward is given for discovering conditions that exhibit a discrepancy between the prediction of the current model and the experimental result. As unexpected observations are incorporated into the model, the reward associated with them vanishes. Results obtained so far indicate that evolutionary search is a useful paradigm for characterizing the phenomenology of context-sensitive components.

1. Introduction

Eugene Wigner once remarked that "[...] the construction of machines, the function of which he can foresee, constitutes the most spectacular accomplishment of the physicist" [27]. The engineering approach in which mental conception precedes physical creation is extremely efficient, but at the same time severely limited by our capacity to predict the behavior of artifacts. If prediction is not feasible, however, engineering often proceeds with an essentially evolutionary approach. The early history of optical engineering is a case in point [11].

Prediction is computation. A physical system can be viewed as computing its own behavior and might be the most effective architecture to do so. Whenever the prediction of the system's behavior is harder than the implementation and evaluation of the system itself, evolution may provide the shortest path to a system design that satisfies given requirements. In a planned design the requirements are used to instruct the construction or modification of the system. Evolution, on the contrary, creates a population of arbitrarily constructed or modified systems and applies the requirements to select a subset of the population for further modification. Thus evolution uses the implementation of a system (parent) to estimate the performance of arbitrarily modified versions of this system (offspring). The evolutionary engineering paradigm is not only capable of coping with self-organizing, globally interacting systems, but is particularly effective in this domain (cf. [24, 26]).

The instructive use of the requirements for systems design is so familiar to human thought, that it is often assumed as the first explanation for any seemingly purposeful design. Nature, however, utilizes evolution's selective paradigm on many different levels of scale. The design of species in ecosystems and of antibodies in the immune system are examples where the initial assumption of an instructive process gave way to an explanation based on a selective process [10]. It appears that conformational state transitions in enzymatic catalysis could be the next process whose explanation will undergo this shift from an instructive to a selective paradigm [17].

The key point that makes evolution attractive from the technological perspective is its reliance on arbitrary rather than directed system modifications. Since evolution does not depend on a predictable relationship between a given modification and system performance it can cope with numerous interacting factors and with situations for which mechanistic models are either not available or so complex as to be impractical [3]. This led to the idea of 'automatic research machines' that can search for ideal parameter settings in real world physical devices [20]. Random changes to the parameters are accepted (or rejected) if the new pa-

rameter settings reduce (or increase) the distance to the design goal.

Evolutionary experimentation thus resembles the approach of a craftsman who does not theorize about his work but nevertheless achieves a high degree of perfection through constant feedback from interaction with the 'hardware' of his craft. The scouting algorithm described below is an attempt to capture another aspect of the craftsman's approach—the curiosity of his mind directing his attention to unexpected observations, free of any particular design goal. The potential significance of the observations can subsequently be contemplated with the aid of human creativity.

1.1. The role of context-sensitive components

The concept of scouting has been developed as a tool to identify behavior of context-sensitive components that may be exploited in computing devices. Components are context-sensitive if their function cannot be described without taking the system which they are part of into consideration. Such components are an invaluable source of complexity for the implementation of pattern processors [30].

Complexity is a computational resource because the algorithmic complexity ($\mathcal{K}$) of the input-output behavior (f) of an information processor can never exceed the combined complexity of the processor's architecture (a) and of the program (p_f) that specifies the behavior of the processor:

$$\mathcal{K}(f) \leq \mathcal{K}(a) + \mathcal{K}(p_f)$$

This follows directly from the definition of algorithmic complexity [4, 13, 25]: $\mathcal{K}(f)$ corresponds to the length of the shortest description sufficient to generate f. Since the program p_f in combination with a machine of architecture a is able to generate f, a description of p_f together with a description of a constitutes a description of f. Hence the shortest way to describe f cannot be longer than the combination of the shortest descriptions of p_f and a. Since almost all possible maps f have high algorithmic complexity $\mathcal{K}(f)$—a consequence of the fact that there are more input-output tables than descriptions that would be shorter than the tables—the low complexity $\mathcal{K}(a)$ of the conventional machine architectures makes programs of formidable complexity $\mathcal{K}(p_f)$ inevitable.

Consider, for example, the simple 10×10 bit pattern (100 bit input) shown in Fig. 1. There are $2^{2^{100}}$ possible classifications of the input patterns into two groups (1 bit output) such as 'accept' and 'reject'. A program that would implement an arbitrary one of these seemingly simple mappings is likely to be so long that it is entirely impractical. Correspondingly, a specialized hardware architecture built from simple components could require a very large number of them. The use of context-sensitive components and their concomitant complex interactions may open up a path for implementing high-complexity pattern processors with significantly less hardware than conventional technologies require.

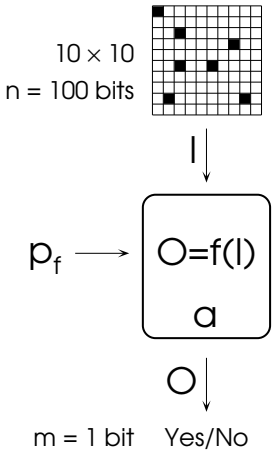

Figure 1. Pattern classification with a general purpose computer. The program p_f has to select the desired classification map f from the set of all possible system behaviors.

Conceivably, nature's molecular information processors employ context-sensitive components to efficiently implement complex computing operations. Moreover, biological systems allow for molecular self-organization to dynamically reconfigure their processing architecture [16].

1.2. Signal processing wetware

A number of different schemes have been proposed to employ specific molecular properties in artificial computing devices. Excitable and oscillating chemical reactions have been considered as a substrate for molecular computing. Rössler showed that many basic computer circuits can be simulated by chemical reaction systems [21]. The possibility of implementing bistable chemical devices has been investigated [23]. Simulation studies show that networks of bistable kinetic devices can exhibit artificial neural network type learning behavior [8] and thus the realization of chemical parallel computing architectures might be practical. For example, Pettit [15] suggested a design in which stacked chemical reactor slices are interfaced through a pumped fluid. That in fact the computational properties of active media can be considered not only in 'paper chemistry', but actually demonstrated in experiments, has been confirmed with a light-sensitive version of the Belousov-Zhabotinsky (cf. [22]) reaction. A thin layer of this reaction medium is capable of performing elementary image processing steps such as contour enhancement and contrast

inversion [12, 18, 19].

In the early 1970s the idea of considering biological structures as natural molecular computers came up [5, 14]. Subsequently numerous schemes for utilizing biomolecules in artificial information processing devices came into consideration (e.g., [1, 2, 9]).

The use of macromolecules for information processing poses a problem in which the sought resource, molecular level interactions, goes hand in hand with the impracticality of foretelling system behavior [6, 7]. This is particularly the case for proteins. As context-sensitive components they are refractory to simple mechanistic modeling but can be characterized empirically [29]. The challenge is to map out component response with respect to numerous interacting factors, and to do so with a limited number of experimental tests.

Fig. 2 shows the computer controlled fluidics of a setup developed to characterize the response of enzymes (i.e., catalytically active proteins) to various milieu conditions. This

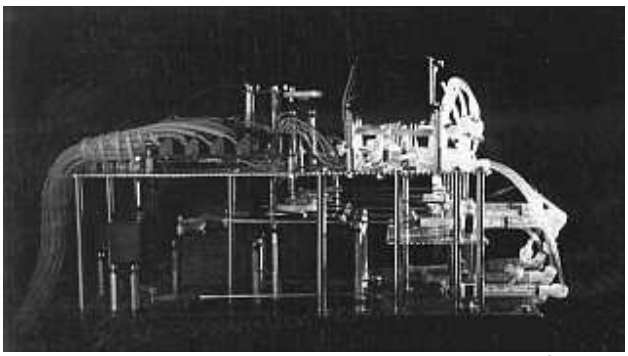

Figure 2. Computer controlled fluidics developed for the automated exploration of protein context-sensitivity. Sixteen servos (of the type used for radio controlled model airplanes) are employed to control 6 syringe pumps, 9 valves and a peristaltic pump (from [28]).

setup enables an evolutionary experimentation algorithm, running on a digital computer, to probe the response of enzymes. The computer can compose a chemical milieu from up to five substances and then measure the catalytic activity of the enzyme in the milieu.

It would be possible to specify a desired signal processing behavior as the goal for the evolutionary experimentation and to use the setup as an 'automatic research machine' of the type proposed by Rechenberg. However, to explore more generally the capabilities of proteins to serve as context-sensitive components in molecular devices, we are currently investigating the scouting algorithm described

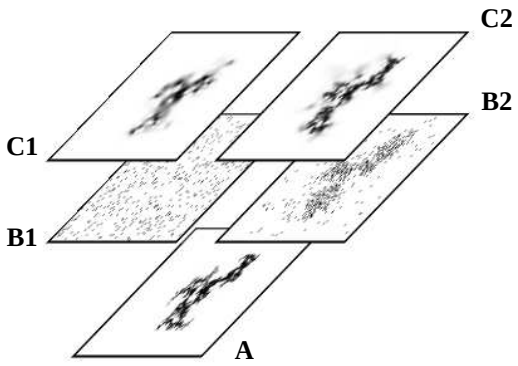

Figure 3. Random sampling vs. scouting. The phenomenon under investigation is represented by a 100×100 pixel gray level image (A). Sampling 5% of A at random locations (B1) leads to estimate C1 of the phenomenon. Exploring A with the scouting algorithm directs the sample locations towards heterogeneous areas and thus can acquire more detailed information with the same number of samples.

in the next section as a method of controlling the fluidics setup.

2. The scouting algorithm

The task of the scouting algorithm is the exploration of complex phenomena. To analyze and illustrate different scouting strategies we decided to emulate phenomena through gray-scale images. The x and y axes correspond to two factors with the gray values in the image plane representing the response for specific factor levels. This method has the advantage that phenomena with arbitrary and complex features can conveniently be prepared. For simplicity the following discussion will assume that only two factors are considered (i.e., a two-dimensional factor space), but all operations extend to higher dimensions as will be required, for example, to drive the five-factor fluidics setup shown in Fig. 2.

Generally in this application the knowledge that can be acquired is limited by the number of experiments that can be conducted, not by available computing resources. It is desirable that the samples be distributed in the factor space for maximum information gain. Fig. 3 illustrates the focusing of the sampling on detail rich areas that occurs with scouting.

The operation of the scouting algorithm that gives rise to the nonuniform sample distribution is depicted in Fig. 4. A population of individuals, each representing a location x

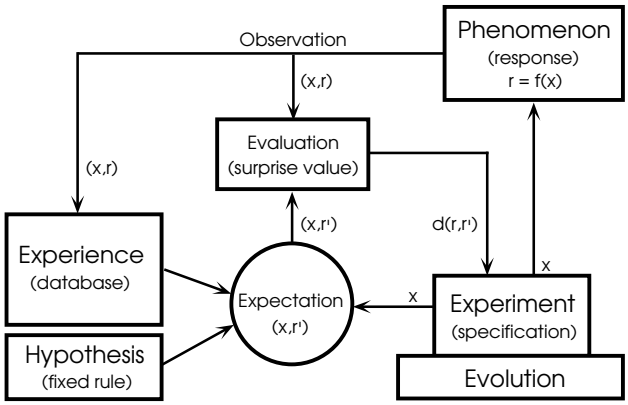

Figure 4. Schematic view of the scouting algorithm. From a population of experiments, the specification (x) of the experiment that gave rise to the most surprising observation (x,r) in view of the current experience is selected for reproduction.

in the factor space, is subjected to evolution. For each individual in the population first an expected response r' is computed from the contents of the experience database using a fixed hypothesis to extrapolate the known experiences. Then the phenomenon is probed for the condition x resulting in a response r. The observation (x, r) is compared to the expectation (x, r'). A large difference $d(r, r')$ between expected and actual response, i.e., a surprising observation, is associated with a high fitness value for the individual that requested the observation. The observation (x, r) is also entered into the database of experiences and can contribute to the evaluation of the next individual. Every experiment performed on the phenomenon contributes to the experience stored in the database and consequently leads to a refinement of the expectation. For all results presented here the population size was ten, all individuals being offspring of the single best individual selected for reproduction. Thus, the individuals live for one generation only. Their fitness would in any case be low after one generation; after they have been evaluated once there is experience available in the database for the factor combination which they represent and subsequent experiments would not yield surprising observations (assuming no measurement errors).

Individuals are derived through random variation of the location in the factor space that the parent individual represents. The magnitude of the variation is a function of the fitness of the parent. (Each factor is varied by a uniformly distributed pseudorandom amount less than ± 0.03 for a parent whose surprise value $d(r, r')$ was above 0.7, less than ± 0.04 if $d(r, r')$ was above 0.2, less than ± 0.2 if $d(r, r')$ was above 0.05, and anywhere within the factor range otherwise.)

It is important to note that the algorithm does not optimize factor levels with regard to some desired response. Instead it rewards those individuals in the population that direct the search towards unexpected response behavior. The expectation is of course not constant and changes as information is acquired.

The expected response for all points in the factor space forms a model of the phenomenon and becomes more refined with each experiment. The factor levels as well as the response level are normalized ($0 \leq x \leq 1$, $0 \leq y \leq 1$, $0 \leq r \leq 1$). The expected response (r') for a location x in the factor space is computed from the observations closest to x. If the database is empty, the expected response for any location in the factor space is set to 0.5 (the middle value of the normalized scale), if only one observation is in the database, the expected response for any x is the same as the one observed response. In general a fixed maximum number of observations r (set to 10 for the runs reported here) is considered in the calculation of an expected response r'.

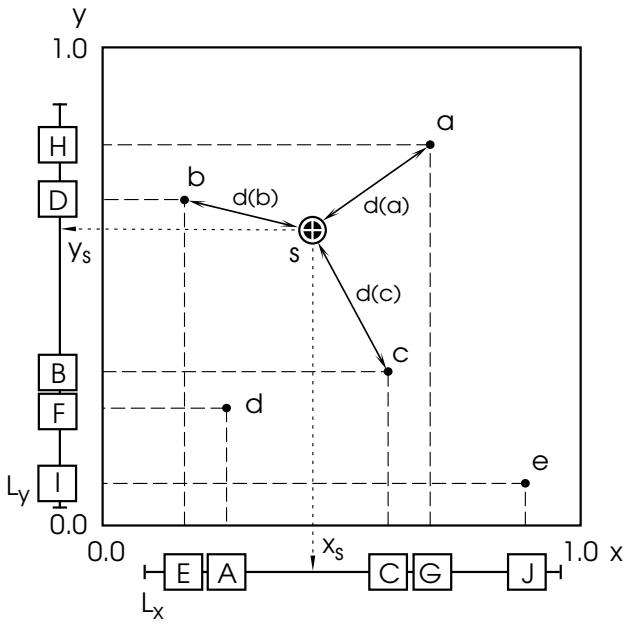

Figure 5. Deriving expectations from experience. The observations (a–e) are each represented in two lists (L_x and L_y) sorted according to the x and y factor levels. The expected response at s is computed from the known responses of a set of observations nearest to s. See text for details.

The observations in the experience database are accessible through lists sorted by the factor levels of the observations. One such list is maintained for each dimension of the

factor space. Fig. 5 illustrates the two-dimensional case, assuming that five observations (labeled a–e in the £gure) have been entered into the experience database. The list L_x represents the five observations sorted according to their x-factor levels and correspondingly L_y represents them sorted according to their y-factor levels. To retrieve the observations relevant to computing the expectation for the specified experiment s=(x_s, y_s), first the lower neighbor of x_s in L_x is accessed (labeled A) and the corresponding observation d retrieved. For this example we assume that only three observations will be used to calculate the expected value. The distance of observation d from s in the factor space is computed and d entered into a list of potentially relevant observations ([d]). The entries in this list are maintained in sorted order with respect to their distance from s. Next, the lower neighbor of s in the next dimension (i.e., from L_y) is considered and consequently observation c is added to the list of potentially relevant observations. Since c is closer to s than d is, it is inserted to the left of d ([c,d]). Subsequently the lists of all dimensions are traversed from the immediate neighbors of the point (x_s, y_s) outward and observations added to the potentially relevant observations. After entry D in list L_y has been considered the list of potentially relevant observations contains [b,c,d].

If all positions in the list of potentially relevant observations are filled, an observation will be sorted into the list if it is closer to s than the farthest observations in the list, the latter then being discarded. If an entry in L_x has an x value that is further from x_s than the rightmost entry in the list of potentially relevant observations, L_x is not further traversed in this direction. Traversal is also stopped at the end of a list. Corresponding rules apply to both directions in all dimensions. The entries remaining in the list of potentially relevant observations after list-traversal has stopped in all dimensions are then used to compute the expectation.

For the data presented we used in all cases the following hypothesis to derive the expectation from the relevant observations. The observed response values are weighted by the cubic distance between the location of the observation and the specified factor space location. The weighted average of the observed response values is then taken as the expected response.

Fig. 6 illustrates the progress of the scouting algorithm's sampling activity. The 100×100 pixel image representing the phenomenon is shown on the top left. It contains two spots of fine detail, that on average are very similar to the rest of the factor space. In rows B–F dots in the left square indicate the locations in the factor space where the phenomenon has been probed. The squares in the right column show the expected gray value for every location in the factor space, given the observations indicated in the left square. In combination they represent an empirical model of the phenomenon.

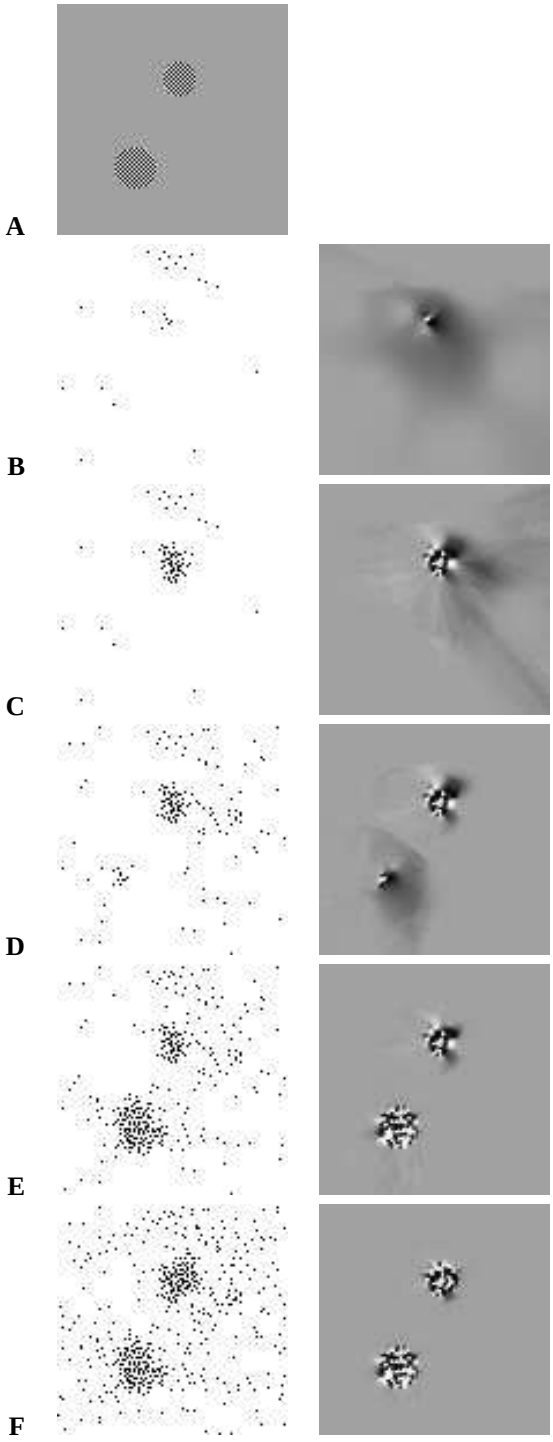

Figure 6. Progress of the sampling. The image representing the phenomenon is shown in A. Sampled locations are shown in the left column, the expected response is shown on the right. B: 25, C: 75, D: 150, E: 325, and F: 500 samples

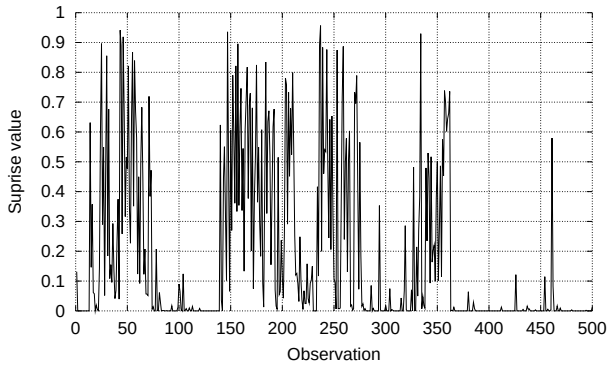

Figure 7. Differences between expected and observed values during the progress of the sampling shown in Fig. 6. The £rst burst corresponds to the sampling of the upper spot, the second burst corresponds to the discovery of the lower spot.

Sampling starts at a random location in the factor space. In the example shown in Fig. 6, the £rst samples are taken in the top center. Because the phenomenon shows a uniform response, only the first sample taken has some surprise value (being different from the 0.5 value assumed in the absence of any observations). Due to the low fitness of the parent selected from the first generation, the offspring shows broad variation in its factor levels and spreads in the factor space. Some individuals sample the area of fine detail in the upper center of the space. Unexpected observations from the factor combinations in this area yield high surprise values (shown in Fig. 7) and correspondingly high fitness for the individuals that discovered the area. The dark shadows visible in the right square of Fig. 6B, i.e., the model derived from the first 25 samples, attest to distant extrapolation in the sparsely sampled space.

As the population concentrates in the high fitness area, the experience in this area increases and consequently the expectations become more accurate. The decrease in surprise value leads to a wider spread in the population and to the possibility that a more interesting area of the space is discovered. After 14 generations the second detailed area in the factor space is discovered (Fig. 6D and the second burst in Fig. 7) and explored.

Fig. 8 shows the clustering of sampling at factor levels that give rise to abrupt response changes and the concomitant reduced sampling density at factor combinations that yield a graded response. Fig. 9 gives an impression of how much the sample clustering varies among different initializations of the pseudorandom generator.

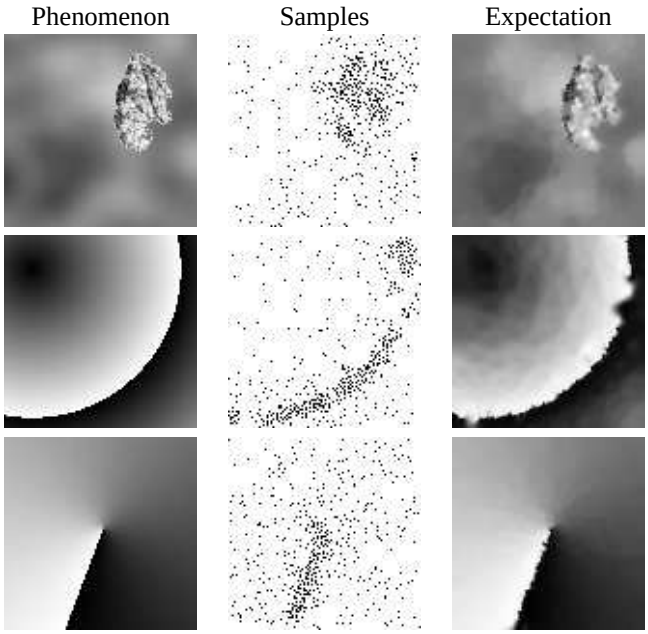

Figure 8. Clustering of 500 sampling locations for different phenomena.

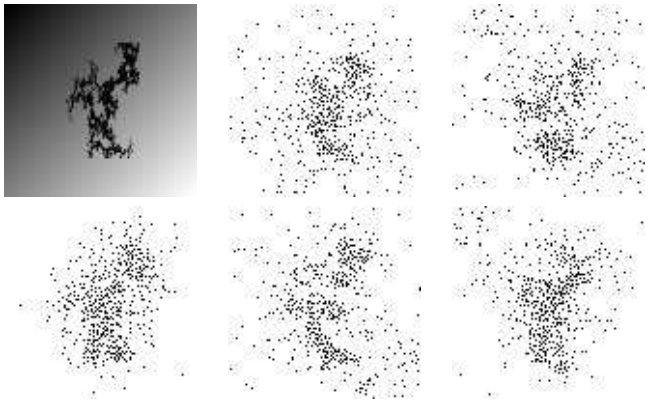

Figure 9. Sample locations of 500 samples for £ve different runs probing the same phenomenon.

3. Concluding remarks

We have developed a simple algorithm that speci£es experiments for sampling the response of an unknown system. The aim is not to build models of context-sensitive systems, but to discover conditions under which the system shows interesting response features. To acquire as much information as possible from a limited number of experiments, it is desirable that the algorithm sample the factor space densely in

areas of complex behavior and sparsely in areas that show less factor interaction.

The results we have obtained so far show that the combination of a very simple strategy for evolutionary experimentation with a simple empirical modeling of the response behavior suf£ces for sample clustering to occur in areas of high interest. These £ndings are based on the application of the algorithm to phenomena emulated by gray-scale images, not to real world experiments. Further tests are required to evaluate the performance of the scouting method under noise in both the response and the factor levels, before it can usefully be applied in actual computer controlled experimentation.

Hardware evolution faces the dichotomy that only real physical implementations can harvest its full potency, but at the same time such implementations are impractical for any system short of a self-replicating one. Methods like scouting may be able to narrow the gap between simulation and experimentation as means for fitness evaluation.

Acknowledgements. We gratefully acknowledge the discussions with our teacher Michael Conrad before his death in December 2000. The reported material is based upon work supported by the U.S. National Science Foundation under Grant Nos. ECS-9704190 and CCR-9610054, and by NASA under Grant No. NCC2-1189, all to Michael Conrad.

References

[1] L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.

[2] T. Aoki, M. Kameyama, and T. Higuchi. Interconnection-free biomolecular computing. *Computer (IEEE)*, 25:41–50, 1992.

[3] G. E. P. Box. Evolutionary operation: A method for increasing industrial productivity. *Appl. Statist.*, 6:81–101, 1957.

[4] G. J. Chaitin. Algorithmic information theory. *IBM Journal of Research and Development*, 21:350–359,496, 1977.

[5] M. Conrad. Information processing in molecular systems. *Currents in Modern Biology (now BioSystems)*, 5:1–14, 1972.

[6] M. Conrad. On design principles for a molecular computer. *Commun. ACM*, 28:464–479, 1985.

[7] M. Conrad and H. M. Hastings. Scale change and the emergence of information processing primitives. *J. theor. Biol.*, 112:741–755, 1985.

[8] A. Hjelmfelt and J. Ross. Mass-coupled chemical systems with computational properties. *J. Phys. Chem.*, 97:7988–7992, 1993.

[9] F. T. Hong. The bacteriorhodopsin model membrane system as a prototype molecular computing element. *BioSystems*, 19:223–236, 1986.

[10] N. K. Jerne. Antibody and learning: Selection versus instruction. In G. C. Quarton, T. Melnechuk, and F. O. Schmitt, editors, *The Neurosciences*, pages 200–205. Rockefeller University Press, New York, 1967.

[11] R. Kingslake. The development of the photographic objective. *J. Optical Society of America*, 24:73–84, 1934.

[12] L. Kuhnert, K. I. Agladze, and V. I. Krinsky. Image processing using light-sensitive chemical waves. *Nature*, 337:244–247, 1989.

[13] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, New York, 2nd edition, 1997.

[14] E. A. Liberman. Cell as a molecular computer (MCC). *Bio£zika*, 17:932–943, 1972.

[15] F. R. Pettit. *Post-Digital Electronics*, chapter 13, pages 161,162,165. Electrical and Electronic Engineering. Ellis Horwood, Chichester, West Sussex, 1982.

[16] J. O. Pfaffmann and M. Conrad. Adaptive information processing in microtubule networks. *BioSystems*, 55:47–58, 2000.

[17] S. D. Rader and D. A. Agard. Conformational substates in enzyme mechanism: the 120K structure of α-lytic protease at 1.5 å resolution. *Protein Sci.*, 6:1375–1386, 1997.

[18] N. G. Rambidi. Nondiscrete biomolecular computing. *Computer (IEEE)*, 25:51–54, 1992.

[19] N. G. Rambidi and A. V. Maximychev. Towards a biomolecular computer: information processing capabilities of biomolecular nonlinear dynamic media. *BioSystems*, 41:195–211, 1997.

[20] I. Rechenberg. Cybernetic solution path of an experimental problem. Technical report, Royal Aircraft Establishment, Library Translation 1122, Farnborough, 1965. Reprinted with commentary in D. B. Fogel, editor, *Evolutionary Computation: The Fossil Record*, pages 297–309, IEEE Press, New York, 1998.

[21] O. E. Rössler. Chemical automata in homogeneous and reaction-diffusion kinetics. In M. Conrad, W. Güttinger, and M. D. Cin, editors, *Physics and Mathematics of the Nervous System*, volume 4 of *Lecture Notes in Biomathematics*, pages 399–418. Springer, Heidelberg, 1974.

[22] S. K. Scott. *Oscillations, waves and chaos in chemical kinetics*. Oxford University Press, Oxford, 1994.

[23] F. F. Seelig and O. E. Rössler. A chemical reaction ¤ip-¤op with one unique switching input. *Zeitschrift für Naturforschung*, 27b:1441–1444, 1972.

[24] J. L. Segovia-Juarez and S. Colombano. Mutation buffering capabilities of the hypernetwork model. In the present volume, 2001.

[25] R. J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7:1–22, 224–254, 1964.

[26] L. G. Volkert and M. Conrad. The role of weak interactions in biological systems: the dual dynamics model. *J. theor. Biol.*, 193:287–306, 1998.

[27] E. P. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Comm. Pure. Appl. Math.*, 13:1–14, 1959.

[28] K.-P. Zauner. *Conformation-based Molecular Computing: Simulation and Implementation*. PhD thesis, Wayne State University, Detroit, Michigan, 2001.

[29] K.-P. Zauner and M. Conrad. Enzymatic pattern processing. *Naturwissenschaften*, 87(8):360–362, 2000.

[30] K.-P. Zauner and M. Conrad. Molecular approach to informal computing. *Soft Computing*, 5(1):39–44, 2001.