Harmonic Competition: A Self-Organizing Multiple Criteria Optimization

Yasuo Matsuyama, *Senior Member, IEEE*

***Abstract*—Harmonic competition is a learning strategy based upon winner-take-all or winner-take-quota with respect to a composite of heterogeneous subcosts. This learning is unsupervised and organizes itself. The subcosts may conflict with each other. Thus, the total learning system realizes a self-organizing multiple criteria optimization. The subcosts are combined additively and multiplicatively using adjusting parameters. For such a total cost, a general successive learning algorithm is derived first. Then, specific problems in the Euclidian space are addressed. Vector quantization with various constraints and traveling salesperson problems are selected as test problems. The former is a typical class of problems where the number of neurons is less than that of the data. The latter is an opposite case. Duality exists in these two classes. In both cases, the combination parameters of the subcosts show wide dynamic ranges in the course of learning. It is possible, however, to decide the parameter control from the structure of the total cost. This method finds a preferred solution from the Pareto optimal set of the multiple object optimization. Controlled mutations motivated by genetic algorithms are proved to be effective in finding near-optimal solutions. All results show significance of the additional constraints and the effectiveness of the dynamic parameter control.**

I. Introduction

COMPETITION is an agent selection mechanism based upon a fitness measure. Given an input, each agent computes the fitness of its state to the input. Upon completion of this stage, every agent broadcasts its own figure of fitness to the others. Then, each agent compares its own fitness with the ones received. An agent which has a better fitness than any of the received values can claim to be the winner.

If the agent state is equivalent to an input weight vector, or if the above structure is well-expressed by a directed graph, the total system is called a competitive artificial neural network. Each agent is called an artificial neuron with competition. "Artificial" is mostly omitted if there is no possibility of mistaking them for wet-ware neurons.

Learning by competition is usually winner-take-all. That is, only the winner obtains the right to produce an output and to learn. Learning means a modification of the state vector so that the fitness to the current input is increased. If there are neurons which cooperate with the winner, these comrades can also learn. This is called winner-take-quota. Hereafter, "state vector," "weight vector," and "neuron" are interchangeably used.

Manuscript received May 1, 1994; revised June 12, 1995. This work was supported in part by the Grant-in-Aid for Scientific Research; Higher-Level Brain Information Processing.

The author is with the Department of Electrical, Electronics, and Computer Engineering, Waseda University, Tokyo 169, Japan. He is also with the Sympat Committee of the RWC Partnership of Japan.

Publisher Item Identifier S 1045-9227(96)01231-3.

The measure of fitness is often called the cost or error. In this case, the winner selection is based upon a minimization. If the cost is an error in the input approximation by weight vectors, the learning is equivalent to a phase of clustering or data compression. If we try to treat more sophisticated or real-world problems, it is necessary to use a composite of heterogeneous subcosts. The most important subcost, the main cost, is for the data approximation. The rest of the subcosts are for various constraints. The competition phase computes the values of fitness for all subcosts. The harmonic competition is a winner selection mechanism taking all heterogeneous subcosts into account.

Incorporation of the aforementioned fitting subcosts in the total error measure has enabled the expansion of the problem class to be solvable by learning. There are several studies treating such additional subcosts [3], [9], [15]–[19], [23], [25]. In these cases, different subcosts are additively combined with the main cost. Since the main cost and subcosts are quite different in nature, combination parameters are essential to adjust the subcosts' dynamic ranges. For instance, the simplest form of such a total cost includes one adjustment parameter, λ, such that $\bar{d} = \bar{f} + \lambda\bar{g}$. Here, $\bar{f}$ is an average cost for the approximation. $\bar{g}$ is a constraint. Except for a few studies, the parameter λ *per se* is fixed throughout the learning. This is meaningful, however, only if a carefully selected value of this parameter is given *a priori*. Since fixing the parameter is part of the design phase, not specifying this number forces ill-conditioned learning. Therefore, designers of the learning system can not set this parameter until many repeated trials are performed: It is necessary to find a reasonable method for dynamically adjusting such a parameter.

Among the above references, [16]–[18] and [23] report the importance of the dynamic control of the subcost. The strategies presented therein, however, were limited to specific problems. It is desirable to find a more universal method. This is one of the main issues in this paper after a formal derivation of general harmonic competition. It is also worth emphasizing that a strategy of controlled mutation of the weight vectors is theoretically derived from the harmonic competition. Thus, the organization of this paper is as follows: First, general derivation of harmonic competition and learning is given. Then, two different classes of problems are studied as benchmarks. These classes show a beautiful duality. The first is the case where the number of neurons, M, is smaller than that of the data, say N. Data compression by divergence-constrained vector quantization (DVQ; all for equiprobability, equierror, and joint equiprobability/error) is such a case. From the competition bias, a strategy of weight mutation is obtained. The second is the opposite case: M is larger than N. A Euclidian traveling salesperson problem (TSP) and extended vehicle routing problems (EVRP) with sets of real-world data are such cases. The EVRP is a case containing all types of constraints. Through general discussions and experiments with the above classes, the following can be claimed:

- Harmonic competition is an eligible strategy to find a preferred solution to the Pareto optimal set (noninferior solution set) [12, pp. 331–332] of the multiple criteria optimization. That is, this learning mechanism picks up a trade-off between the main cost and subcosts.
- The parameter λ has quite a wide dynamic range. Setting this parameter requires some sort of prior knowledge. Therefore, the subcosts should be controlled so that their ratio to the main cost changes slowly. This adjustment covers a wide dynamic range of the parameter λ. It gives a better, or at least, a comparable performance to the static method which relies upon *a priori* information.
- The weight mutation can be derived from the competition bias. This method helps to avoid bad local minima.
- The rule presented is applicable to the case of multiple constraints. A wide variety of problems can be cast in the learning algorithm presented.

II. Cost with Penalties

As was introduced in Section I, harmonic competition is a phase of finding a learning neuron for the optimization of the total cost. The total cost is made up of two parts: the main cost and subcosts. The main cost has the role of measuring the degree of data approximation. The subcosts are selected according to the nature of the problem to be solved. Ubiquitous applications use only the main cost. In what follows, however, we present a class of composite costs whose usage will widely expand the class of problems solvable by competitive learning. The harmonic cost, i.e., the total cost, is the key concept of the following discussions. The terminology harmonic competition refers to competitive learning of the harmonic cost.

A. Combination of Subcosts

Let $\{\boldsymbol{x}_n\}_{n=0}^{N-1}$ be a set of input vectors to the learning network which contains a set of weight vectors $\{\boldsymbol{w}_m\}_{m=0}^{M-1}$. Let $Q(\boldsymbol{x},\boldsymbol{w})$ be a mapping such that

$$Q(\boldsymbol{x},\boldsymbol{w}) = \begin{cases} 1, & \text{if the weight vector } \boldsymbol{w} \text{ is assigned} \\ & \text{to the input vector } \boldsymbol{x}\,, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The average cost of the approximation by the weight vectors is then

$$\bar{f} = \frac{1}{N}\sum_{n=0}^{N-1} f(\boldsymbol{x}_n,\boldsymbol{w}_m)Q(\boldsymbol{x}_n,\boldsymbol{w}_m) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=0}^{N-1} f_n Q(\boldsymbol{x}_n,\boldsymbol{w}_m). \tag{2}$$

This is the main cost. Let $g_{nk}(\{\boldsymbol{x}_i\}_{i=0}^{N-1},\{\boldsymbol{w}_j\}_{j=0}^{M-1})$, $(k = 0,\cdots,K-1)$, be constraints added to f_n. Let $h_{n\ell}$ $(\{\boldsymbol{x}_i\}_{i=0}^{N-1}$, $\{\boldsymbol{w}_j\}_{j=0}^{M-1})$, $(\ell = 0, \cdots, L-1)$, be multiplicative penalties. The cost of the mapping $Q(\boldsymbol{x}_n,\boldsymbol{w}_m)$ is then

$$D_n = \sum_{m=0}^{M-1}\left(f_n + \sum_{k=0}^{K-1}\lambda_{nk}g_{nk}\right)\left(\prod_{\ell=0}^{L-1}h_{n\ell}\right)Q(\boldsymbol{x}_n,\boldsymbol{w}_m) \stackrel{\text{def}}{=} N d_n. \tag{3}$$

Here, $\boldsymbol{x}_n = \boldsymbol{x}^{(t)}$ is the actual input at the tth data supply of the learning phase. That is, the superscript t is the number of iterations treated as time count. On the other hand, $\tau = \lfloor t/N \rfloor$ is called the sweep. The parameters λ_{nk}, $(k = 0,\cdots,K-1)$, combine the subcosts. These parameters appear when a convex multiple criteria optimization is transformed into a cost function approach. The average of the total cost, i.e., the harmonic cost, is then

$$\bar{d} = \sum_{n=0}^{N-1} d_n = \frac{1}{N}\sum_{n=0}^{N-1} D_n. \tag{4}$$

The first step of harmonic competition is to find a minimization element for d_n with respect to a successively given input data $\boldsymbol{x}^{(t)} = \boldsymbol{x}_n$

$$\boldsymbol{w}_{m(n)}^{(t)} = \arg\min_{0\le m<M} d_n = \arg\min_{0\le m<M} D_n. \tag{5}$$

Ties are broken appropriately. The above $\boldsymbol{w}_{m(n)}^{(t)}$ is called the winner which is treated as the case where $Q(\boldsymbol{x}_n,\boldsymbol{w}_{m(n)}^{(t)}) = 1$. The neural weight vector is updated for learning by

$$\boldsymbol{w}_{m(n)}^{(t+1)} = \boldsymbol{w}_{m(n)}^{(t)} + \Delta\boldsymbol{w}_{m(n)}^{(t)} \tag{6}$$

where the modification term is

$$\Delta\boldsymbol{w}_{m(n)}^{(t)} = -\frac{\varepsilon^{(t)}N}{2}\frac{\partial d_n}{\partial \boldsymbol{w}_{m(n)}^{(t)}} = -\frac{\varepsilon^{(t)}}{2}\frac{\partial D_n}{\partial \boldsymbol{w}_{m(n)}^{(t)}}. \tag{7}$$

$\varepsilon^{(t)}$ is a learning parameter. The derivation in (7) is

$$\begin{aligned} N\frac{\partial d_n}{\partial \boldsymbol{w}_{m(n)}^{(t)}} &= \frac{\partial D_n}{\partial \boldsymbol{w}_{m(n)}^{(t)}} \\ &= \left(\frac{\partial f_n}{\partial \boldsymbol{w}_{m(n)}^{(t)}} + \sum_{k=0}^{K-1}\lambda_{nk}^{(t)}\frac{\partial g_{nk}}{\partial \boldsymbol{w}_{m(n)}^{(t)}}\right)\left(\prod_{\ell=0}^{L-1}h_{n\ell}\right) \\ &\quad + \left(f_n + \sum_{k=0}^{K-1}\lambda_{nk}^{(t)}g_{nk}\right)\left\{\frac{\partial}{\partial \boldsymbol{w}_{m(n)}^{(t)}}\left(\prod_{\ell=0}^{L-1}h_{n\ell}\right)\right\}. \end{aligned} \tag{8}$$

If we consider a cooperative neighborhood, $\mathcal{N}_{m(n)}^{(t)}$, of the winner $\boldsymbol{w}_{m(n)}^{(t)}$, then the following update is also applied:

$$\boldsymbol{w}_{\mathcal{N}\{m(n)\}}^{(t+1)} = \boldsymbol{w}_{\mathcal{N}\{m(n)\}}^{(t)} + \alpha_{\mathcal{N}\{m(n)\}}^{(t)}\Delta\boldsymbol{w}_{m(n)}^{(t)}. \tag{9}$$

Here, the parameter $\alpha_{\mathcal{N}\{m(n)\}}^{(t)}$ specifies a degree of cooperation, possibly affected by the multiplicative handicap $h_{n\ell}$ [15], [17].

The above learning strategy has the following effective interpretation: The harmonic competition by (5) and (6) indicates only the most appropriate agent with respect to the input is eligible to learn. This means that the learning is undertaken by the agent which suffices to change the total system only the least. In this sense, the harmonic competition is an embodiment of the minimum learning principle.

B. Weight Update for the Euclidian Cost

Here, we compute the case where the term f_n of (3) is Euclidian. Note that the class with additive subcosts is already significantly wide. Thus, this case is discussed first. Multiplicative penalties will be discussed later jointly with the extended vehicle routing problems.

Consider the case where $\lambda_{nk} \stackrel{\text{def}}{=} \lambda$ and $h_{n\ell} \equiv 1$; i.e., $K = L = 1$ and the combination parameters are independent of n. Then

$$D_n = \sum_{m=0}^{M-1} \left(\|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 + \lambda g_n\right) Q(\boldsymbol{x}_n, \boldsymbol{w}_m).$$

Here, $\|\cdot\|$ is the Euclidian metric. Then, one obtains from (4) that

$$\begin{aligned}\bar{d} &= \frac{1}{N}\sum_{n=0}^{N-1} D_n \\ &= \frac{1}{N}\sum_{n=0}^{N-1} \|\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}\|^2 + \lambda \frac{1}{N}\sum_{n=0}^{N-1}\left\{\sum_{m=0}^{M-1} g_n Q(\boldsymbol{x}_n, \boldsymbol{w}_m)\right\} \\ &= \frac{1}{N}\sum_{n=0}^{N-1} \|\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}\|^2 + \lambda \frac{1}{N}\sum_{n=0}^{N-1} [g]_{m(n)} \\ &\stackrel{\text{def}}{=} \bar{f} + \lambda \bar{g} \end{aligned} \tag{10}$$

where $[g]_{m(n)}$ means that the winner $\boldsymbol{w}_{m(n)}$ is identified in g_n. Then, one obtains from (7) and (8) that

$$\Delta \boldsymbol{w}_{m(n)}^{(t)} = \varepsilon^{(t)}\left(\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}^{(t)}\right) - \frac{\varepsilon^{(t)}\lambda^{(t)}}{2}\frac{\partial [g]_{m(n)}}{\partial \boldsymbol{w}_{m(n)}^{(t)}}. \tag{11}$$

Here, the superscript t for $\varepsilon^{(t)}$ and $\lambda^{(t)}$ specifies the values at the tth data input. Thus, the harmonic competitive learning is a combination of actions described by (5), (6), (9), and (11). The derivation of the winner $\boldsymbol{w}_{m(n)}^{(t)}$ in (10) requires detailed specifications of $\bar{g}$. Therefore, this part is discussed in Sections III and IV depending upon the form of $\bar{g}$.

The control of the parameters $\varepsilon^{(t)}$ and $\lambda^{(t)}$ also depends on the class of problems. The learning parameter $\varepsilon^{(t)}$ can be controlled by

$$\varepsilon^{(t+1)} = \varepsilon^{(t)} + \Delta\varepsilon(\bar{f}^{(t)}, \bar{f}^{(t-1)}, \bar{g}^{(t)}, \bar{g}^{(t-1)}, t).$$

This form includes the case of predefined control. We note here that $\varepsilon^{(t)}$ need not monotonically decrease to zero. In some regularization problems, a monotone increase with saturation is even desirable.

In any problem, the subcost parameter $\lambda^{(t)}$ can be fixed provided an appropriate value is known *a priori* (static rule). This figure, however, can never be given in advance. Thus, the problem itself is ill conditioned. One naive method is to perform repeated experiments changing this value. Since the main cost and the subcost are heterogeneous in nature, many experiments are required to cover the wide dynamic range of λ. Another method is to use a dynamic rule

$$\lambda^{(t+1)} = \lambda^{(t)} + \Delta\lambda(\bar{f}^{(t)}, \bar{f}^{(t-1)}, \bar{g}^{(t)}, \bar{g}^{(t-1)}, t).$$

One will find that this is a method to pick up a preferred solution from the Pereto optimal set using the harmonic competition dynamics.

Starting from Section III, specific strategies of the harmonic competition are given. Treated problems and their significances are as follows:

- The case where the number of neurons, M, is smaller than that of the data, N, (data compression): Divergence is selected as the subcost for the minimization. Effective competition biases and strategies are obtained in the following cases:
  - a) Divergence-constrained vector quantization for equiprobability.
  - b) Divergence-constrained vector quantization for equierror.
  - c) Divergence-constrained vector quantization for joint equiprobability and equierror.
- The case of $M > N$ (regularization).
  - a) Euclidian traveling salesperson problem.
  - b) Extended vehicle routing problem (an example of three subcosts with multiplicative constraints).

The duality of the above two cases is an important subject to be observed throughout the text.

III. Data Compression with Constraints

Data compression is the case in which the number of neurons, M, is less than that of the data, N. In this case, $Q(\boldsymbol{x}_n, \boldsymbol{w}_m) = 1$ means that the input data $\boldsymbol{x}_n$ is approximated by the weight vector $\boldsymbol{w}_m$. Then, each input is expressed by $\log_2 M$ bits achieving data compression. Since $\boldsymbol{x}_n$ is a vector, such a case is called vector quantization. Since the input vectors are fed into the learning mechanism one by one, the strategy is called a successive mode. On the other hand, there is a learning method which uses the whole training data set repeatedly [11]. This is called a batch mode. Both successive and batch modes of their plain versions suffer from traps at undesirable local minima. We present methods which use the divergence (Kullback–Leibler number) to make up this deficiency. The divergence can be effective either for output probability equalization (equiprobability) or for output error equalization (equierror).

A. Equiprobability Vector Quantization

The equiprobability constraint is incorporated as a subcost in the case where the usage of the weight vectors $\boldsymbol{w}_m$, $(m = 0, \cdots, M-1)$, is requested to be uniform. This means that the output entropy is maximized, or equivalently, the output divergence is minimized. Since the divergence includes a target probability, nonuniform distributions can be the design object. In the experiments, only uniform distribution is treated since there is no specific requirement for nonuniformity in applications so far.

Let p_m be the probability that the weight vector $\boldsymbol{w}_m$ is selected. Then, p_m is the expectation of Q; $p_m = \mathcal{E}[Q(\boldsymbol{x}_n, \boldsymbol{w}_m)]$. Denote $\{p_m\}_{m=0}^{M-1}$ by $\mathcal{P}$. Let $\mathcal{Q} \stackrel{\text{def}}{=} \{q_m\}_{m=0}^{M-1}$

be a set of desirable output probabilities; e.g., $q_m = 1/M$ for the uniform case. The divergence is then

$$\bar{g}(\mathcal{P}\|\mathcal{Q}) = \sum_{m=0}^{M-1} p_m \log\frac{p_m}{q_m}.$$

Here, log is the natural logarithm. The average cost (10) for equiprobability harmonic competition is then

$$\bar{d} = \bar{f} + \lambda\bar{g} = \frac{1}{N}\sum_{n=0}^{N-1} \|\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}\|^2 + \lambda \sum_{m=0}^{M-1} p_m \log\frac{p_m}{q_m}.$$

That is

$$\bar{d}=\frac{1}{N}\sum_{n=0}^{N-1}\left\{\sum_{m=0}^{M-1}\|\boldsymbol{x}_n-\boldsymbol{w}_m\|^2 Q(\boldsymbol{x}_n,\boldsymbol{w}_m)\right\}+\lambda\sum_{m=0}^{M-1} p_m \log\frac{p_m}{q_m}.$$

Since the data $\boldsymbol{x}_n$ is drawn uniformly from the finite source $\{\boldsymbol{x}_n\}_{n=0}^{N-1}$, the probability p_m in the long run is as follows:

$$p_m = \frac{1}{N}\sum_{n=0}^{N-1} Q(\boldsymbol{x}_n, \boldsymbol{w}_m).$$

Then, one obtains

$$\begin{aligned}\bar{d} &= \frac{1}{N}\sum_{n=0}^{N-1}\left\{\sum_{m=0}^{M-1}\left(\|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 + \lambda\log\frac{p_m}{q_m}\right)Q(\boldsymbol{x}_n,\boldsymbol{w}_m)\right\}\\ &= \frac{1}{N}\sum_{n=0}^{N-1}\left\{\|\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}\|^2 + \lambda\log\frac{p_{m(n)}}{q_{m(n)}}\right\}.\end{aligned}$$

That is

$$[g]_{m(n)} = \log\frac{p_{m(n)}}{q_{m(n)}}. \tag{12}$$

Thus, harmonic competition for the equiprobability is

$$\boldsymbol{w}_{m(n)}^{(t)} = \arg\min_{0\le m<M}\left\{\|\boldsymbol{x}_n - \boldsymbol{w}_m^{(t)}\|^2 + \lambda\log\frac{p_m}{q_m}\right\}. \tag{13}$$

Note that (12) is the log-conscience [16] whose approximation of $\log x \approx x-1$ around $x=1$, as well as, $q_m = 1/M$ is the conscience [5]. The second term, $\Delta\boldsymbol{w}_{m(n)}^{(t)}$, of (11) can be omitted in successive data inputs. Thus, the increment is

$$\Delta\boldsymbol{w}_{m(n)}^{(t)} \approx \varepsilon^{(t)}(\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}^{(t)}). \tag{14}$$

The learning algorithm is then described as follows.

Equiprobability Harmonic Competition:

Step 1) (initialization; $t = 0$)

A set of training data $\{\boldsymbol{x}_n\}_{n=0}^{N-1}$ and a set of initial weight vectors $\{\boldsymbol{w}_m^{(0)}\}_{m=0}^{M-1}$ are given.

Step 2) (data feeding; increment t)

A data $\boldsymbol{x}_n$ is given. The harmonic competition then selects the winner $\boldsymbol{w}_{m(n)}^{(t)}$ using (13).

Step 3) (weight update)

Update the winner's weight $\boldsymbol{w}_{m(n)}^{(t)}$ using (6) and (14).

Step 4) (self-organization; optional)

Update the cooperating neurons using (9) and (14).

Step 5) (test and termination)

If a predefined stop condition is met (e.g., the number of iterations), then iteration is halted. Otherwise, modify $\varepsilon^{(t)}$, $\lambda^{(t)}$ and $\alpha_{\mathcal{N}\{m(n)\}}^{(t)}$ according to the given rules. Then, go to Step 2).

From both computational and performance view points, Step 5) includes additional strategies.

Sweep-Based Update; Computational Cost Reduction: In Step 5), the quantity of the competition bias (p_m in the case of equiprobability) is computed only at the end of every sweep. Since the update of the parameters $\varepsilon^{(t)}$, $\lambda^{(t)}$ and $\alpha_{\mathcal{N}\{m(n)\}}^{(t)}$ is related to this competition bias, the sweep count is used as the time index. That is, $\varepsilon^{(t)}$, $\lambda^{(t)}$ and $\alpha_{\mathcal{N}\{m(n)\}}^{(t)}$ are kept constant for $N\tau \le t < (\tau+1)N$.

Additional Strategies for Increasing Performance: In Step 5), additional strategies can be incorporated to increase the total performance. Dynamic split and weight vector mutation are such strategies. Explicit descriptions of these methods will be given later.

B. Equierror Vector Quantization

Equierror vector quantization is considered to be asymptotically optimal for errors in the form of difference distortion measures [8, p. 376], [24]. One should, however, use this property only when there is a considerable number of weight vectors for much more rich source data. This is because the property holds only in the limit. Since vector quantization is used for low to medium rate compression, exact equierror should not be requested. It is expected, however, that the optimal or near-optimal solutions appear as almost equierror. This is a clue in finding a strategy for the escape from bad local minima where the distribution of errors is uneven.

Let r_m be a normalized subtotal of errors due to usage of the weight vector $\boldsymbol{w}_m$. That is

$$r_m = \bar{f}_m/\bar{f}$$

with

$$\begin{cases}\bar{f}_m = \dfrac{1}{N}\sum_{n=0}^{N-1}\|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 Q(\boldsymbol{x}_n,\boldsymbol{w}_m),\\ \bar{f} \;= \sum_{m=0}^{M-1}\bar{f}_m.\end{cases}$$

Therefore, $\sum_{m=0}^{M-1} r_m = 1$ holds. Let $\mathcal{R} = \{r_m\}_{m=0}^{M-1}$. The subcost for the equierror is then

$$\bar{g} = D(\mathcal{R}\|\mathcal{Q}) = \sum_{m=0}^{M-1} r_m \log\frac{r_m}{q_m}.$$

The total cost is then

$$\begin{aligned}\bar{d} &= \bar{f} + \lambda\bar{g}\\ &= \frac{1}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 + \lambda\sum_{m=0}^{M-1} r_m\log\frac{r_m}{q_m}\\ &= \frac{1}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 Q(\boldsymbol{x}_n,\boldsymbol{w}_m)\\ &\quad+\frac{\lambda}{N\bar{f}}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 Q(\boldsymbol{x}_n,\boldsymbol{w}_m)\log\frac{r_m}{q_m}\\ &= \frac{1}{N}\sum_{n=0}^{N-1}\left[\sum_{m=0}^{M-1}\left\{\left(1+\frac{\lambda}{\bar{f}}\log\frac{r_m}{q_m}\right)\|\boldsymbol{x}_n - w_m\|^2\right\}Q(\boldsymbol{x}_n,\boldsymbol{w}_m)\right].\end{aligned}$$

That is

$$[g]_{m(n)} = \frac{\|\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}\|^2}{\bar{f}} \log \frac{r_{m(n)}}{q_{m(n)}}.$$

Then, the harmonic competition for the equierror is

$$\boldsymbol{w}_{m(n)}^{(t)} = \arg \min_{0 \le m < M} \left\{ \left(1 + \frac{\lambda}{\bar{f}} \log \frac{r_m}{q_m} \right) \|\boldsymbol{x}_n - \boldsymbol{w}_m^{(t)}\|^2 \right\}. \quad (15)$$

Thus, the log-conscience acts as a multiplicative penalty. This corresponds to a shunting inhibition [13]. For the weight update, (6), (9), and (14) are used.

The equierror harmonic competition is described as follows.

Equierror Harmonic Competition: Replace Step 2) of the equiprobability harmonic competition as follows.

Step 2) (data feeding; increment t)

A data $\boldsymbol{x}_n$ is given. The harmonic competition then selects the winner $\boldsymbol{w}_{m(n)}^{(t)}$ using (15).

In the sweep-based update of Step 5), p_m is replaced by r_m.

C. Vector Quantization Jointly with Equiprobaility and Equierror

The constraint on both equiprobability and equierror is expected to show stronger effects on the exiting from bad local minima.

Let u_m be a normalized subtotal of joint probability and errors with respect to usage of the weight vector $\boldsymbol{w}_m$. That is

$$u_m = \frac{p_m r_m}{\sum_{\ell=0}^{M-1} p_\ell r_\ell} \stackrel{\text{def}}{=} \frac{\bar{\eta}_m}{\bar{\eta}}.$$

In this case, $\sum_m u_m = 1$. Let

$$\bar{g} = D(\mathcal{U} \| \mathcal{Q}) = \sum_{m=0}^{M-1} u_m \log \frac{u_m}{q_m}.$$

Then, the total cost is

$$\begin{aligned}
\bar{d} &= \bar{f} + \lambda \bar{g} \\
&= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 Q(\boldsymbol{x}_n, \boldsymbol{w}_m) \\
&\quad + \lambda \sum_{m=0}^{M-1} u_m \log \frac{u_m}{q_m} \\
&= \frac{1}{N} \sum_{n=0}^{N-1} \left[\sum_{m=0}^{M-1} \left\{ \left(1 + \frac{\lambda p_m}{\bar{\eta}} \log \frac{u_m}{q_m} \right) \|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 \right. \right. \\
&\quad \left. \left. Q(\boldsymbol{x}_n, \boldsymbol{w}_m) \right\} \right].
\end{aligned}$$

That is

$$[g]_{m(n)} = \frac{p_m \|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2}{\bar{\eta}} \log \frac{u_m}{q_m}.$$

This is again a multiplicative penalty. Thus, the harmonic competition for the joint equiprobability/error is

$$\boldsymbol{w}_{m(n)}^{(t)} = \arg \min_{0 \le m < M} \left\{ \left(1 + \frac{\lambda p_m}{\bar{\eta}} \log \frac{u_m}{q_m} \right) \|\boldsymbol{x}_n - \boldsymbol{w}_m^{(t)}\|^2 \right\}. \quad (16)$$

The joint equiprobability/error competitive learning is described as follows.

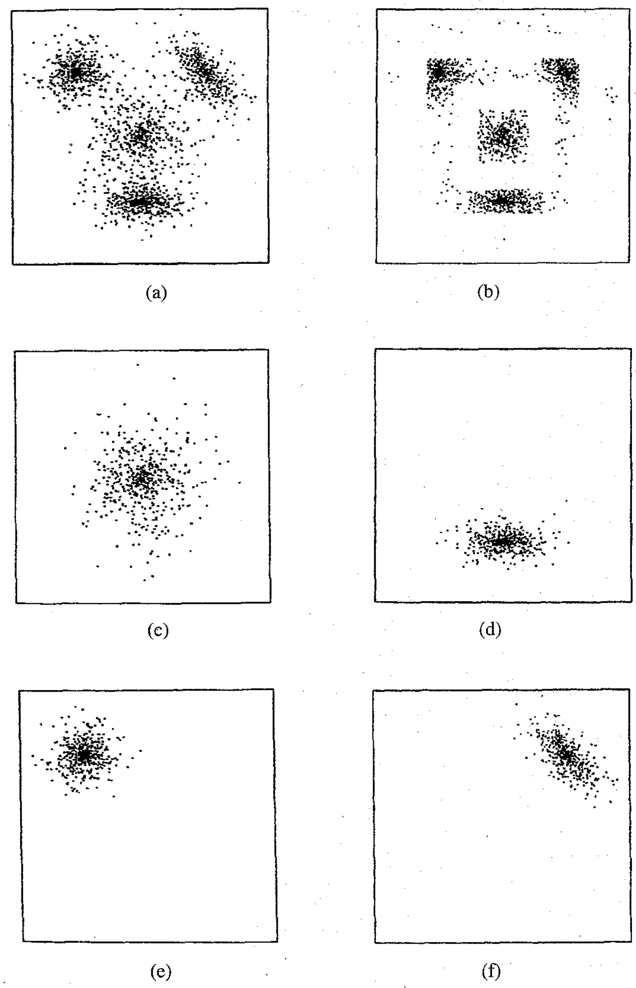


Fig. 1. Generation of training data. (a) Data set A. (b) Data set B. (c)–(f) Four Gaussian clusters used for the generation of data set A.

Joint Equiprobability/Error Harmonic Competition: Replace Step 2) of the equiprobability harmonic competition as follows.

Step 2) (data feeding; increment t).

A data $\boldsymbol{x}_n$ is given. Harmonic competition then selects the winner $\boldsymbol{w}_{m(n)}^{(t)}$ using (16).

In the sweep-based update of Step 5), p_m is replaced by u_m.

D. Experiments of Equiprobability, Equierror, and Joint Equiprobability/Error

In all of the following vector quantization experiments, the updating of cooperative neurons [Step 4)] was not performed. This is because our target was the minimization of the approximation error taking the exit from bad local minima into account. In the experiments of Section IV, however, updating of the cooperating neurons [Step 4)] will be identified as an important phase of learning.

1) Two Sets of Training Data: First, we prepare two sets of training data illustrated in Fig. 1(a) and (b). Set A is a simple mixture of four Gaussian clusters in a unit square.

- A_1: $(\mu_x, \mu_y) = (0.5, 0.5), (\sigma_x, \sigma_y) = (0.15, 0.15); 512$ points.
- A_2: $(\mu_x, \mu_y) = (0.5, 0.75), (\sigma_x, \sigma_y) = (0.1, 0.005); 512$ points.
- A_3: $(\mu_x, \mu_y) = (0.25, 0.25), (\sigma_x, \sigma_y) = (0.0075, 0.0075); 512$ points.
- A_4: $(\mu_x, \mu_y) = (0.75, 0.25), (\sigma_x, \sigma_y) = (0.1, 0.05); 512$ points with a rotation of $-45°$.

These are shown in Fig. 1(c)–(f). The total number of training data is $N = 2048$. The number of weight vectors is $M = 32$.

Set A looks complicated, however, it is rather well natured. We prepare another set for establishing the presented learning. Set B has two square doughnut ditches centered at (0.5,0.5) with a width of 0.1. The exterior corners of the ditches include (0.1,0.1) and (0.3,0.3). The total number of data is kept to be $N = 2048 = 512 \times 4$. In set B, there are a few isolated points. In plain successive learning, neurons trapped at such points can never move anywhere. This occurs even if the neighborhood update of Step 4) is used.

2) Control of the Learning Parameter $\varepsilon^{(t)}$: The learning parameter $\varepsilon^{(t)}$ necessarily tends to zero as t becomes large. This does not mean, however, that $\varepsilon^{(t)}$ is monotonically decreasing. There are occasions where it is better for $\varepsilon^{(t)}$ to increase. We choose the dynamic control of $\varepsilon^{(t)}$ to be updated at every sweep (N data supplies). That is, the sweep-based update is adopted in Step 5) of the learning algorithms. This is because of the reduction in the neuron's communication cost. Let the sweep index be $\tau = \lfloor t/N \rfloor$. The rule is as follows:

$$\varepsilon^{(\tau+1)} = \max\{\varepsilon^{(\tau)} + \Delta\varepsilon^{(\tau)}, \varepsilon^{\max}\}. \tag{17}$$

Here

$$\Delta\varepsilon^{(\tau)} = \begin{cases} \gamma_\varepsilon(\bar{f}^{(\tau-1)} - \bar{f}^{(\tau)})/\bar{f}^{(\tau-1)}, & \text{if } \bar{f}^{(\tau)} < \bar{f}^{(\tau-1)} \\ & \text{and } \bar{g}^{(\tau)} > \bar{g}^{(\tau-1)}, \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

with

$$\gamma_\varepsilon = \begin{cases} 0.5, & \text{if } \varepsilon^{(\tau+1)} < \varepsilon^{(\tau)}, \\ 0.25, & \text{if } \varepsilon^{(\tau+1)} \geq \varepsilon^{(\tau)}. \end{cases} \tag{19}$$

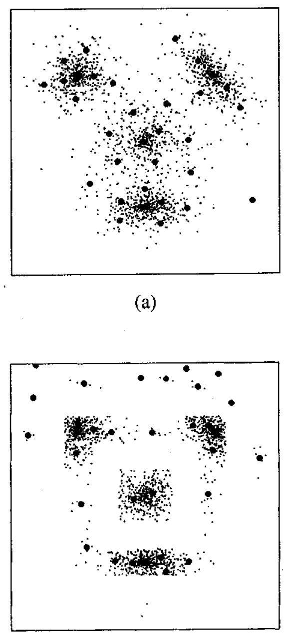
The control (18) allows the learning parameter $\varepsilon^{(\tau)}$ to increase, while the control (19) ensures the decreasing trend of $\varepsilon^{(\tau)}$.

3) Dynamic Split: Our initial experiment is to see how ill-natured the data sets A and B are. We start with the case where the initial weight vectors are decided by random numbers. The next experiment with an additional strategy is the aforementioned dynamic split of the weight vectors which is incorporated in Step 5). The strategy of splitting the weight vectors has been widely accepted in the batch mode where all N data are exposed to the learning mechanism simultaneously. Our purpose here is to see how effective the splitting is in the successive learning.

Weight Vector Splitting for Successive Learning: First, one weight vector is placed at the centroid of the training set. After every κ sweeps (κN data supplies), the positions of the weight vectors are copied. Both the original and copied weight vectors may be perturbed by the addition of small random numbers. After $\kappa \log_2 M$ sweeps, the positions of the M weight vectors are decided.

TABLE I
APPROXIMATION ERROR $N\bar{f}$ FOR PLAIN STRATEGIES ($\lambda = 0$)

Strategy	Set A	Set B
Successive learning started with random weights	2.883	2.314
Successive learning with dynamic split	2.839	1.769
Batch learning after dynamic split	2.922	1.790



(a)

(b)

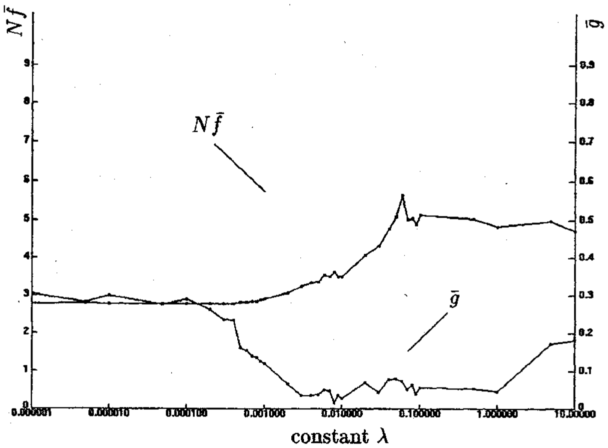
Fig. 2. Learning results starting with a random initial weight vector set: Plain strategy ($\lambda = 0$). (a) Resulting weight vectors for data set A. (b) Resulting weight vectors for data set B.

The dynamic split is regarded as a process of finding a good initial state. Thus, one may use this weight vector set as the initial one for learning. Table I summarizes the effect of the dynamic split with $\kappa = 10$ sweeps by comparing the approximation error $N\bar{f}$. The results of the batch mode after the dynamic split are also given. For both of data sets A and B, the dynamic split is found to be quite effective although the results are still suboptimal. Observe that the batch mode is more likely to get captured at inferior local minima. Fig. 2 shows the results of successive learning starting with random weights. The result of set A is almost all right. It is still, however, at a local minimum containing a null neuron to be eliminated. With the result of set B, one easily realizes it is a terrible pattern. These patterns of the resulting weight vectors will be compared later with those of the improved strategies.

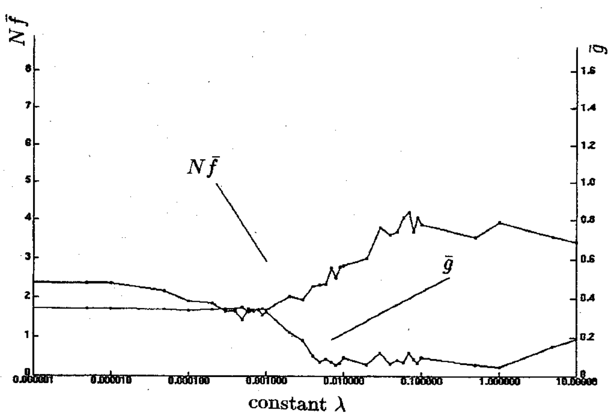
4) Effects of the Subcosts for a Fixed λ: Here, we show the effects of the subcosts $D(\mathcal{P}\|\mathcal{Q})$, $D(\mathcal{R}\|\mathcal{Q})$ and $D(\mathcal{U}\|\mathcal{Q})$. Fig. 3(a) and (b) are the results of equiprobability constraint

TABLE II
MINIMUM VALUES OBTAINED FOR A FIXED λ WITH DYNAMIC SPLIT

Strategy	Set A			Set B		
	λ	$N\bar{f}$	$\bar{g}$	λ	$N\bar{f}$	$\bar{g}$
Equiprobability	0.000001	2.783	0.3042	0.0006	1.640	0.3449
Equierror	0.0002	2.772	0.0400	0.0002	1.552	0.1028
Equipr/error	0.0050	2.791	0.5330	1.0000	1.610	0.3168



(a)

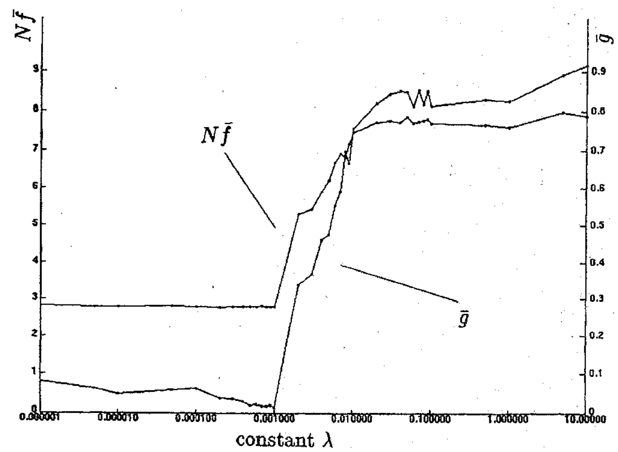


(b)

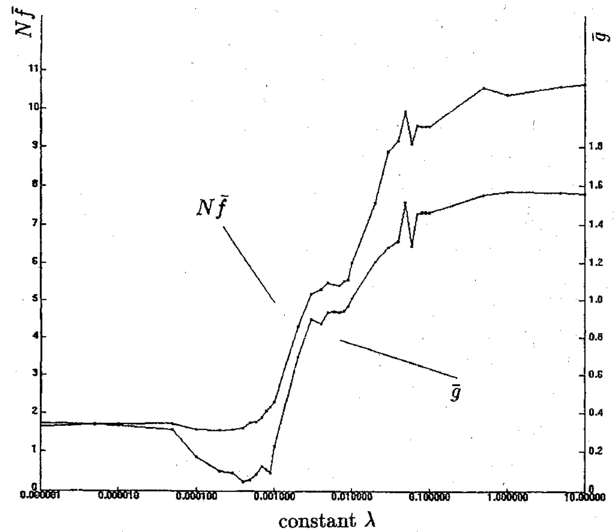
Fig. 3. Performance of log-equiprobability competition with constant λ. (a) Result on data set A. (b) Result on data set B.

by $D(\mathcal{P}\|\mathcal{Q})$ with the parameter λ fixed. Fig. 4(a) and (b) are the results of equierror constraint by $D(\mathcal{R}\|\mathcal{Q})$. Fig. 5(a) and (b) are the results of joint equiprobability/error constraint by $D(\mathcal{U}\|\mathcal{Q})$. In all experiments, the result of the dynamic split is used as the initial weights. The vertical axes specify the approximation error $N\bar{f}$ and each $\bar{g}$. The horizontal axes specify the combination parameter λ in the logarithmic scale. Table II summarizes the minimum values for a fixed λ. These figures and table show the followings clearly:

- All figures indicate that excessive constraint by a large λ degrades the performance of the approximation $N\bar{f}$.
- Fig. 3 indicates that the equiprobability by minimizing $\bar{f}$ and $\bar{g}$ conflict with each other. There is a trade-off relationship.



(a)



(b)

Fig. 4. Performance of log-equierror competition with a constant λ. (a) Result on data set A. (b) Result on data set B.

- The equierror constraint of Fig. 4 indicates that minimizing $\bar{f}$ and $\bar{g}$ is compatible. Small $\bar{f}$ (actually, small $N\bar{f}$ in the figure) is achieved at a small value of $\bar{g}$.
- Fig. 5 shows that the constraint of the joint equiprobability/error is quite strong. The approximation performance $N\bar{f}$ is flat over a wide range of the combination parameter λ. This is a desirable property. The minimum value, however, is larger than that in the case of equierror (see Table II).
- In all cases, the dynamic range of λ is quite wide. Appropriate values of λ are never given *a priori* (see Table II).
- The minimum value of the approximation error is given by the equierror harmonic competition.

5) Dynamic Control of λ: The minimization of $\bar{f}$ and $\bar{g}$ is a typical case of multiple criteria optimization. Usually, the solution is not a unique pair $(\bar{f}, \bar{g})$. Actually, every λ possesses a corresponding point in the Pareto optimal set which is a family of extreme pairs of $\bar{f}$ and $\bar{g}$. Thus, the determination

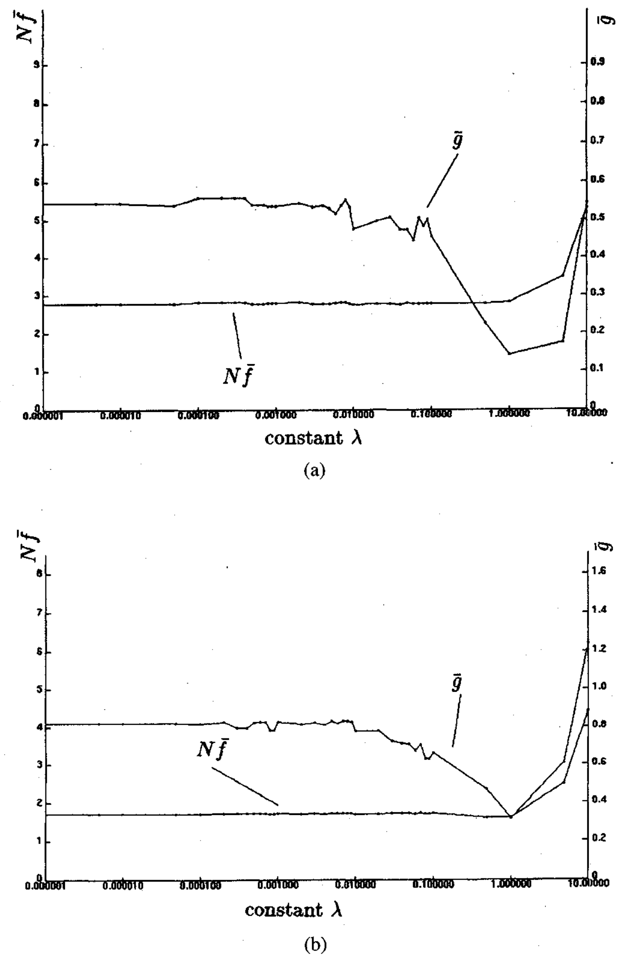


Fig. 5. Performance of log-equiprobability/error competition with constant λ. (a) Result on data set A. (b) Result on data set B.

of an appropriate λ to find a preferred solution requires some more conditions. Since our purpose is to find the minimal $\bar{f}$, a decreasing trend of λ is preferred. Direct control of λ, however, may not be stable because of its wide dynamic range (see Figs. 3–5). Considering the relationship

$$\bar{d} = \bar{f} + \lambda\bar{g} = \bar{f}(1 + \lambda\bar{g}/\bar{f}) \stackrel{\text{def}}{=} \bar{f}(1 + \bar{\mu})$$

we adjust the parameter $\lambda^{(\tau)}$ by the subcost ratio control

$$\mu^{(\tau)} = \lambda^{(\tau)}\bar{g}^{(\tau)}/\bar{f}^{(\tau)} = (1 + a\tau)^{-1}. \tag{20}$$

Note that the superscript τ stands for the sweep ($\tau = \lfloor t/N \rfloor$). Thus, the controlled parameter λ is updated only for every sweep as was explained in Section III-A. This is because of the reduction in the communication cost. The coefficient a can easily be chosen by taking the maximum number of iterations into account.

Table III summarizes the results of the dynamic control of λ using the control (20). Both equiprobability and equierror are tried. Fig. 6(a)–(g) describes the resulting weight vector positions and the learning process for the case of equierror using (20) ($a = 0.14$). In Fig. 6(c)–(g), the horizontal axes specify the number of sweeps τ. From Table III, the followings are noted:

TABLE III
PERFORMANCE ACHIEVED BY A DYNAMICALLY CONTROLLED λ

Strategy	Set A	($a = 0.14$)	Set B	($a = 0.03$)
	$N\bar{f}$	$\bar{g}$	$N\bar{f}$	$\bar{g}$
Equiprobability	2.898	0.1176	1.720	0.1929
Equierror	2.787	0.0500	1.570	0.1737

- The equierror competition is usually superior to the equiprobability with respect to the approximation error $N\bar{f}$.
- The resulting approximation errors for the equierror are close to the smallest values given in Table II. Such values can be obtained by a wide range of the coefficient a.

Fig. 6 indicates the following:

- In Fig. 6(a), the regular positioning of the weight vectors can be observed according to the shape of the data clusters.
- As in Fig. 6(d), the learning parameter $\varepsilon^{(\tau)}$ can go up, however, the trend is to decrease.
- As in Fig. 6(e), the wide dynamic range of $\lambda^{(\tau)}$ is covered.

Instead of using the dynamic control (20), there is a perfect autonomous control of $\lambda^{(\tau)}$

$$\mu^{(\tau+1)} = \mu^{(\tau)} + \Delta\mu^{(\tau)}. \tag{21}$$

The amount $\Delta\mu^{(\tau)}$ is described by observing the rise and fall of $\bar{f}^{(\tau)}$, $\bar{f}^{(\tau-1)}$, $\bar{g}^{(\tau)}$ and $\bar{g}^{(\tau-1)}$. The rule is similar to (18) and (19). The performance is similar to that of case (20). The computation is more complex, therefore, we omit the details here.

6) Log-Conscience Mutation of the Weight Vectors: So far, we have observed that a combination of strategies (dynamic initial split, log-equierror competition and dynamic control of λ) was the best way to guide the learning process to an almost optimal result. In this section, we consider a method of guiding the performance to a near-optimal solution in a discontinuous way. The method is log-conscience mutation. We choose the case of log-equierror for the following explanation: That is, $\{r_m\}_{m=0}^{M-1}$ is selected. In the case of other conscience mechanisms, r_m is replaced by p_m or u_m.

Log-Conscience Mutation for Equierror:

Step 1)

Let $\{r_m\}_{m=0}^{M-1}$ be sorted. That is, $r_i \le r_j$ for $i < j$, as well as $r_0 = r_{\min}$ and $r_{M-1} = r_{\max}$.

Step 2)

Compute

$$z_m = M\left\{\left(\log \frac{r_m}{r_{\min}}\right) \Big/ \left(\sum_{i=0}^{M-1} \log \frac{r_i}{r_{\min}}\right)\right\}. \tag{22}$$

Note that $\sum_{m=0}^{M-1} z_m = M$. The real number set $\{z_m\}_{m=0}^{M-1}$ is rounded to a set of integers $\{s_m\}_{m=0}^{M-1}$. Fractions are rounded to one or zero according to their magnitudes so that $\sum_{m=0}^{M-1} s_m = M$ is maintained.

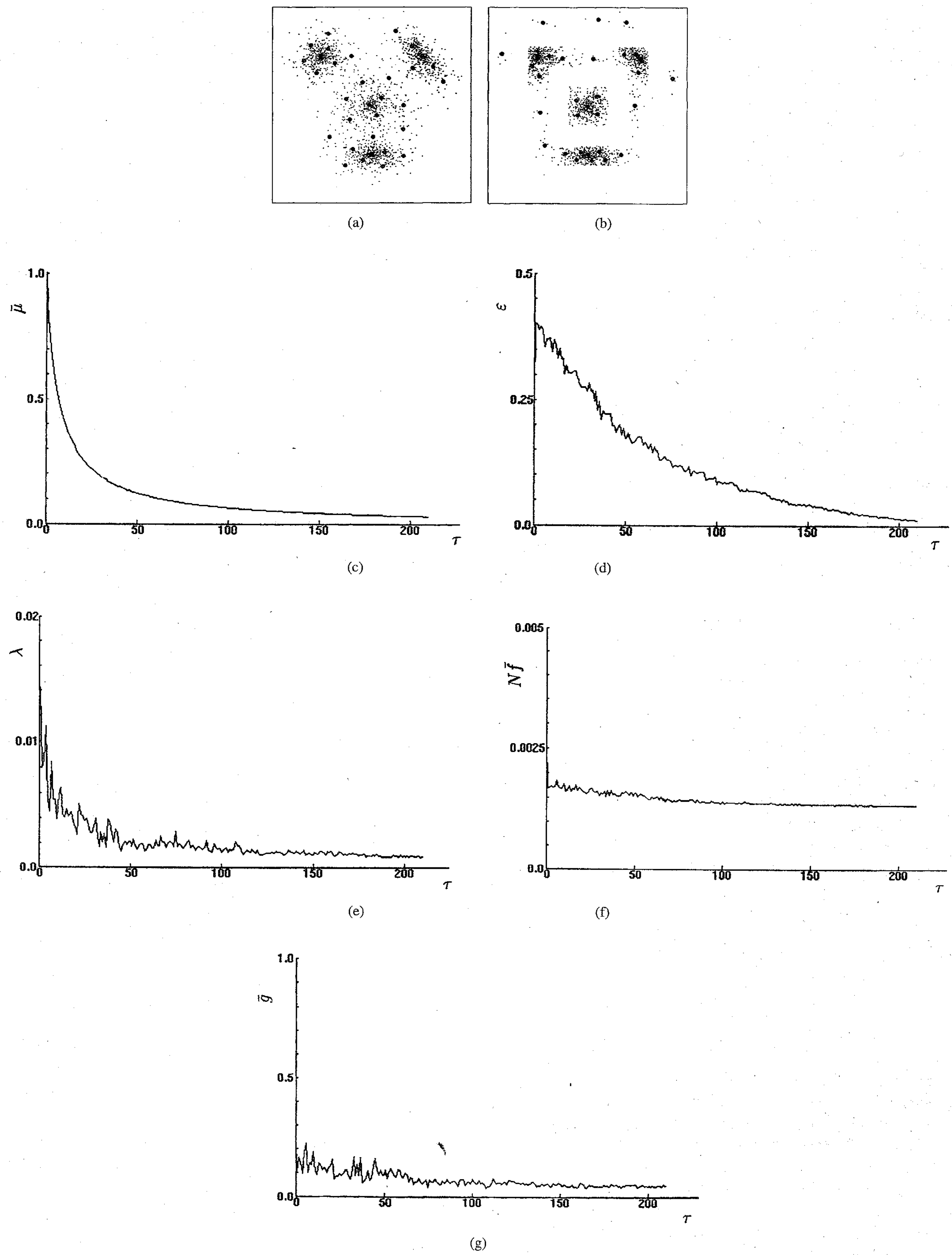


Fig. 6. Resulting weight vector sets by log-equierror with a dynamically controlled λ. (a) Result on data set A. (b) Result on data set B. (c) Specified adjustment of $\bar{\mu}$. (d) Trend of ε. (e) Dynamic control of λ obtained. (f) Convergence of the performance $N\bar{f}$. (g) Effective constraint $\bar{g}$.

TABLE IV
EFFECT OF A LOG-CONSCIENCE MUTATION AFTER DYNAMIC SPLIT

Strategy	Set A			Set B		
	λ	$N\bar{f}$	$\bar{g}$	λ	$N\bar{f}$	$\bar{g}$
Log-equipr mutation	0.000005	2.764	0.2736	0.00005	1.539	0.3269
Log-equierror mutation	0.01	2.774	0.0400	0.009	1.539	0.0952
Log-equipr/error mutation	0.09	2.778	0.3512	0.008	1.544	0.5974

1) log-equipr mutation with $\lambda = 0.000005$ is a result of intensive search (see Fig. 7).

Step 3)
There are three cases

$$\begin{cases} s_m \geq 2 & w_m \text{ is copied } (s_m - 1) \text{ times onto} \\ & \quad \text{other } w_n \text{ 's for which } s_n = 0 \text{ ,} \\ s_m = 1 & \text{no operation on } w_m \text{ ,} \\ s_m = 0 & w_m \text{ is changed to one} \\ & \quad \text{of other } w_n \text{ 's for which } s_n \geq 2 \text{ .} \end{cases} \tag{23}$$

The log-conscience mutation of (23) maintains $\sum_{m=0}^{M-1} s_m = M$. In the weight copy of Step 3), some small noise may be added as was seen in the dynamic split.

For the cases of equiprobability and joint equiprobability/error, the logarithmic quantity $\log(r_m/r_{\min})$ is replaced by $\log(p_m/p_{\min})$ and $\log(u_m/u_{\min})$, respectively.

It is important to point out that the above mutation strategy is based on the log-consciences (13), (15), and (16), respectively. The log-conscience terms result from the original harmonic competition of the cost (10). Thus, the log-conscience always exists in each competition. The mutation is interpreted as occasional strong applications of the log-conscience. In [22], a similar mechanism to our log-conscience mutation is used. Their method, however, is simply based on r_m^b. The exponent $b < 1$ is an experimentally determined *ad hoc* number. Thus, there is no theoretical foundation for the method of equierror learning. Besides, there is no competition-bias at all. Taking these into account, their case corresponds to $\lambda^{(\tau)} \equiv 0$ and r_m^b, $(b < 1)$, for the approximation of our $\log(r_m/r_{\min})$. Since $\lambda^{(\tau)} \equiv 0$ and the mutation should be ceased before convergence, their method stays within the bounds of the selection of better initial weight vectors.

Fig. 7(a) and (b) shows the results of log-equiprobability mutation with dynamic split. Fig. 8(a) and (b) shows the results of log-equierror mutation with dynamic split. Table IV compares the minimum values of the approximation errors $N\bar{f}$ for the three strategies; log-equiprobability, log-equierror, and joint log-equiprobability/error with mutations. From these figures and table, one finds the following:

- Fig. 7(a) and (b) indicate that log-equiprobability mutation is again incompatible with the approximation performance.
- From Fig. 8(a) and (b), one observes that the approximation error $N\bar{f}$ is flat over a wide rage of the combination parameter λ. Its level is good enough (compare with Fig. 4). This is quite a desirable property.
- From Table IV together with Fig. 7(a) and (b), and Fig. 8(a) and (b), the log-equierror mutation after dynamic split is found to be the best. Comparing the approximation errors of Table IV with those of Tables II and III, one finds most of the minimum values here. Equalization of the error by logarithmic order was proved to be effective.

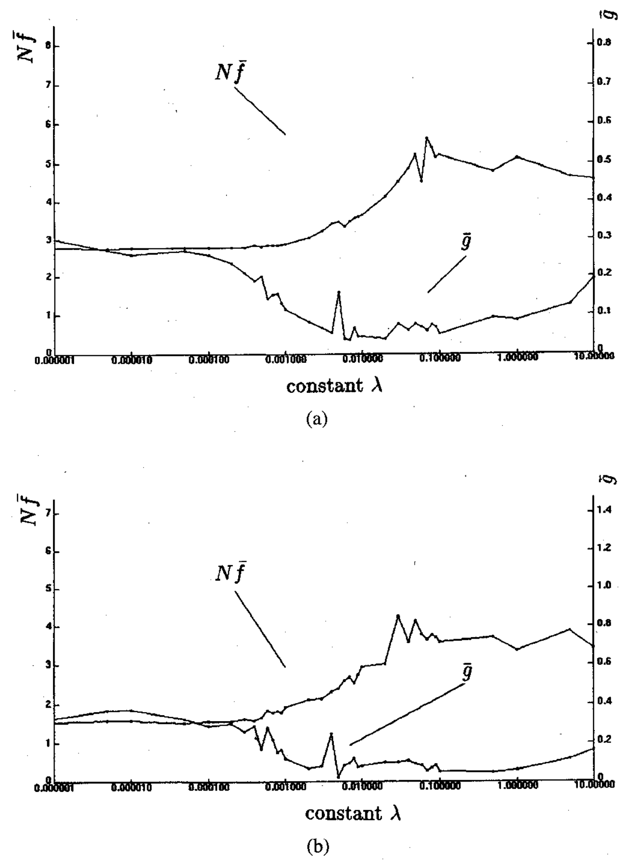


Fig. 7. Performance of log-equiprobability competition with dynamic split, constant λ and mutation. (a) Result on data set A. (b) Result on data set B.

Next, we measure basic ability of the log-equierror mutation without the dynamic split of initial weights. Table V is the resulting performance starting with the same random initial weight set used in the case of Fig. 2. The performance of the log-equierror is again around the minimum obtained. It is slightly inferior to the values given in Table IV, however, where the initial dynamic split exists. Fig. 9 (a) and (b) are the resulting positions of the weight vectors for log-equierror started with the random set. Comparing Fig. 9 with Fig. 2, we can easily observe quite an improvement. Null neurons never appear. The similarity of Fig. 6 and Fig. 9 can also be observed.

In the last experiment, we measure the case of the full strategy: dynamic split, dynamic control of λ and log-equierror mutation. Table VI shows the performance. Comparing Table VI with Tables II–V, we can judge that the full strategy (dynamic split, dynamic control of λ, and log-equierror mutation) is not necessary. Thus, too strict an equierror property is harmful.

7) Recommended Strategies: We have presented the following five strategies:

a) Dynamic control of the learning parameter $\varepsilon^{(\tau)}$.
b) Dynamic split of the initial weight vectors.

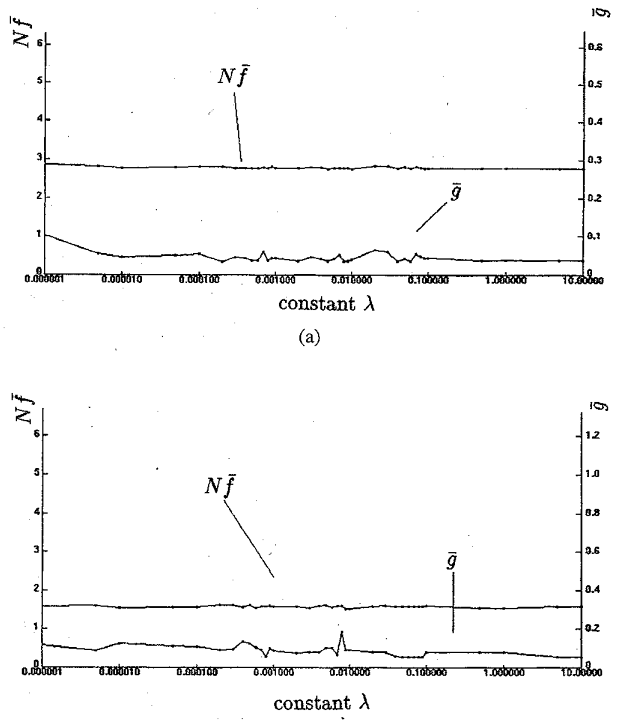


Fig. 8. Performance of log-equierror competition with dynamic split, constant λ and mutation. (a) Result on data set A. (b) Result on data set B.

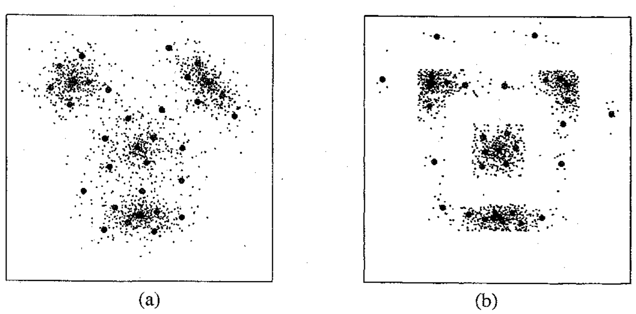


Fig. 9. Resulting weight vector set by log-equierror mutation started with the random set ($\lambda = 0.001$). (a) Result on data set A. (b) Result on data set B.

c) Static existence of log-conscience with a fixed combination parameter λ (equiprobability, equierror, joint equiprobability/error).
d) Dynamic control of the combination parameter λ (equiprobability, equierror, joint equiprobability/error).
e) Weight vector mutation (equiprobability, equierror, joint equiprobability/error).

The dynamic control of $\varepsilon^{(\tau)}$, i.e., strategy a) mainly improves the convergence speed. It may be replaced by a monotone decreasing function. Strategy b), the dynamic split, is always recommended. The static λ of c) is easy to apply; however, it requires *a priori* knowledge of the number. If this value is not possible to know in advance, repeated experiments

TABLE V
EFFECT OF A LOG-CONSCIENCE MUTATION STARTING WITH RANDOM WEIGHTS

Strategy	Set A		Set B	
	$N\bar{f}$	$\bar{g}$	$N\bar{f}$	$\bar{g}$
Log-equipr mutation ($\lambda = 0.000$)	2.848	0.2479	1.595	0.3388
Log-equipr mutation ($\lambda = 0.001$)	2.915	0.1403	1.883	0.1221
Log-equierror mutation ($\lambda = 0.000$)	2.815	0.0598	1.584	0.0344
Log-equierror mutation ($\lambda = 0.001$)	2.773	0.0425	1.561	0.1321
Log-equierror mutation ($\lambda = 0.010$)	2.805	0.0433	1.598	0.0790
Log-equierr/pr mutation ($\lambda = 0.001$)	2.861	0.6234	1.559	0.5484

TABLE VI
EFFECT OF THE FULL STRATEGY

Strategy	Set A		Set B	
	$N\bar{f}$	$\bar{g}$	$N\bar{f}$	$\bar{g}$
Log-equipr bias and mutation	2.889	0.2529	2.144	0.1094
Log-equierror bias and mutation	2.819	0.0505	1.609	0.0802

like Fig. 3–5 are required. Such computationally demanding occasions are rather usual. The dynamic control of d) or the weight vector mutation of e) is preferable. In that case, log-conscience should be used for the equierror. Thus, our recommendation based upon the approximation performances is listed as follows:

1) $\left[\text{b) + x (log-equierror)}\left\{\text{c) + e) with a fixed small}\lambda\right\}\right]$.
2) $\left[\text{b) + (log-equierror)d)}\right]$.
3) $\left[\text{(log-equierror)}\left\{\text{c) + e) with a fixed small}\lambda\right\}\right]$.
4) $\left[\text{b) alone}\right]$.

Note that the strategy "b) alone" is recommended only if running the risk of local optima is allowed.

E. α-divergence and log-AEP

Divergence has a more general form [2], [10]. It is called α-divergence (see the equation shown at the bottom of the next page.) We used the case of $\alpha = -1$ on the normalized measures $\mathcal{P}$ and $\mathcal{Q}$ (Kullback–Leibler number). The choice of α affects the degree of the log-conscience and mutation. That is, this number can change the trend of the logarithmic asymptotic equipartition property (AEP) in the sense of the quantization for source coding. The case of $\alpha = -1$ gave our *log-AEP* in probability, error, and joint probability/error. There exists a general trend that the effect becomes weaker as α increases [18]. The optimal value of α, however, depends on both the nature of the source data and the number of neurons. Thus, the optimal choices for the data set A and B are different. Such a fine tuning is not required so far.

IV. REGULARIZATION WITH CONSTRAINTS

Here, we discuss a class of problems with the number, M, of neurons which is larger than that of the data, N. Since the set of data is fed into a larger population of neurons, additional constraints are necessary. This is called regularization which prevents ill-conditioning. Among such types, Euclidian traveling salesperson problems (Euclidian TSP) and extended vehicle routing problems (extended VRP) are addressed here. The traveling salesperson problem via harmonic competition is a typical case of $M > N$. The extended vehicle routing problems are much more complicated than TSP. They are good examples of the multiple criteria case.

A. Euclidian Traveling Salesperson Problem

1) Formulation and Algorithm: The traveling salesperson problem is a well-known NP-complete problem [7]. If the set of cities visited by a salesperson is located in a Euclidian space, the problem is described as follows.

Euclidian TSP: Given a positive number, T, and a set of cities, $\boldsymbol{X} = \{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{M-1}\}$ specified by position vectors, $\boldsymbol{x}_m \in R^L(\forall m)$, determine if there exists an ordering $(\boldsymbol{x}_{\pi(0)}, \cdots, \boldsymbol{x}_{\pi(M-1)})$ which satisfies

$$\sum_{m=0 \bmod M}^{M-1} d\big(\boldsymbol{x}_{\pi(m)}, \boldsymbol{x}_{\pi(m+1)}\big) \leq T. \tag{24}$$

Here, d is the Euclidian metric and π is a permutation.

This problem is equivalent to finding the minimum value of the total tour length with respect to the permutation π. We are interested in finding good approximations to the minimal length tour using competitive learning. Two-dimensional cases ($L = 2$) are of special interest.

Let

$$\bar{f} = \frac{1}{N}\sum_{n=0}^{N-1} \|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2$$

and

$$\begin{aligned}\bar{g} &= \sum_{m=0\,(\mathrm{mod}M)}^{m-1} \|\boldsymbol{w}_m - \boldsymbol{w}_{m+1}\|^2 \\ &= \sum_{j=m(n)\,(\mathrm{mod}M)}^{m(n)+M-1} \|\boldsymbol{w}_j - \boldsymbol{w}_{j+1}\|^2 = [g]_{m(n)}.\end{aligned}$$

The use of the square norm is for simplicity of the learning equations. The difference from the case of the Euclidian metric is absorbed by the change in learning parameters. The total cost is

$$\bar{d} = \frac{1}{N}\sum_{n=0}^{N-1} D_n = \bar{f} + \lambda\bar{g}$$

with

$$D_n = \sum_{m=0}^{M-1}\left\{\|\boldsymbol{x}_n - \boldsymbol{w}_m\|^2 + \lambda \sum_{j=0(\mathrm{mod}M)}^{M-1} \|\boldsymbol{w}_j - \boldsymbol{w}_{j+1}\|^2\right\} Q(\boldsymbol{x}_n, \boldsymbol{w}_m).$$

Therefore, the harmonic competition is

$$\boldsymbol{w}_{m(n)}^{(t)} = \arg\min_{0\leq m<M} D_n = \arg\min_{0\leq m<M} \|\boldsymbol{x}_n - \boldsymbol{w}_m^{(t)}\|^2.$$

The update term is obtained from (7) as follows:

$$\begin{aligned}\Delta\boldsymbol{w}_{m(n)}^{(t)} &= \varepsilon^{(t)}\left\{\boldsymbol{x}_n - \boldsymbol{w}_{m(n)}^{(t)}\right\} \\ &\quad + \alpha^{(t)}\left\{\boldsymbol{w}_{m(n)-1}^{(t)} - 2\boldsymbol{w}_{m(n)}^{(t)} + \boldsymbol{w}_{m(n)+1}^{(t)}\right\}\end{aligned} \tag{25}$$

where $\alpha^{(t)} \overset{\text{def}}{=} \varepsilon^{(t)}\lambda^{(t)}$. An important point of this problem is how to adjust the parameters $\varepsilon^{(t)}$ and $\lambda^{(t)}$, or equivalently, $\varepsilon^{(t)}$ and $\alpha^{(t)}$. In the previous study [17], the temporal change in these numbers was predefined. In this paper, dynamic control of these parameters is attempted.

The rule of control of the learning parameter is as follows: The update is computed at every sweep (N data supplies). The superscript τ is used instead of t. That is, $\tau = \lfloor t/N \rfloor$. The learning parameter $\varepsilon^{(\tau)}$ is adjusted by the following:

$$\varepsilon^{(\tau)} = \max\{\varepsilon^{(\tau)} + \Delta\varepsilon^{(\tau)}, \varepsilon^{\max}\} \tag{26}$$

with

$$\Delta\varepsilon^{(\tau)} = \begin{cases} 0, & \text{if } \bar{f}^{(\tau)} > \bar{f}^{(\tau-1)} \text{ and } \bar{g}^{(\tau)} < \bar{g}^{(\tau-1)}, \\ \gamma_\varepsilon \dfrac{\bar{f}^{(\tau-1)} - \bar{f}^{(\tau)}}{\bar{f}^{(\tau-1)}}, & \text{otherwise.} \end{cases} \tag{27}$$

The increment γ_ε is

$$\gamma_\varepsilon = \begin{cases} 0.5, & \text{if } \varepsilon^{(\tau+1)} < \varepsilon^{(\tau)}, \\ 0.25, & \text{if } \varepsilon^{(\tau+1)} \geq \varepsilon^{(\tau)}. \end{cases} \tag{28}$$

Comparing the (26)–(28) with (17)–(19), one finds a duality. Equations (26)–(28) ensure the increasing trend of the learning parameter $\varepsilon^{(\tau)}$. On the other hand, (17)–(19) show the decreasing trend of $\varepsilon^{(\tau)}$.

The first adjustment rule for $\lambda^{(\tau)}$ is based upon the following subcost ratio:

$$\mu^{(\tau)} = \lambda^{(\tau)}\bar{g}^{(\tau)}/\bar{f}^{(\tau)} \tag{29}$$

$$D^{(\alpha)}(\mathcal{P}\|\mathcal{Q}) = \begin{cases} \displaystyle\sum_{m=0}^{M-1}\left(q_m - p_m + p_m \log\frac{p_m}{q_m}\right), & (\alpha = -1) \\ \displaystyle\frac{4}{1-\alpha^2}\sum_{m=0}^{M-1}\left\{\frac{1-\alpha}{2}p_m + \frac{1+\alpha}{2}q_m - p^{(1-\alpha)/2}q_m^{(1+\alpha)/2}\right\}, & (-1 < \alpha < 1) \\ D^{(-1)}(\mathcal{Q}\|\mathcal{P}), & (\alpha = 1). \end{cases}$$

with

$$\mu^{(\tau)} = a\tau. \tag{30}$$

This is again a dual case of (20). The second adjustment rule is a weaker version on the rate of the growth

$$\mu^{(\tau)} = a\tau/\varepsilon^{(\tau)}. \tag{31}$$

The third rule for adjusting $\lambda^{(\tau)}$ is the following:

$$\mu^{(\tau+1)} = \mu^{(\tau)} + \Delta\mu^{(\tau)} \tag{32}$$

with (33) shown at the bottom of the page. Here

$$\bar{\eta} = \bar{g}^{(\tau)}/\bar{f}^{(\tau)}. \tag{34}$$

This is an autonomous control rule. In later experiments, the above three adjustment rules on $\lambda^{(\tau)}$ will be examined.

The harmonic competition for TSP is summarized as follows.

Harmonic Competition for TSP with Dynamic Adjustment:

Step 1) (initialization; $t = 0$, $\tau = \lfloor t/N \rfloor = 0$)

The following data set and initial values are given:

- Set of cities $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=0}^{N-1}$.
- Set of neural weight vectors $\boldsymbol{W} = \{\boldsymbol{w}_m\}_{m=0}^{M-1}$. Here, each vector $\boldsymbol{w}_m$ is placed on a closed curve.
- A dynamic rule for $\varepsilon^{(\tau)}$ of (26).
- A dynamic rule for $\alpha^{(\tau)} = \varepsilon^{(\tau)}\lambda^{(\tau)}$ by one of the (30)–(32).
- Neighborhood weight $f(m,\tau)I(|m| < h^{(\tau)})$. Here, $I(\mathcal{B})$ is an $\{0,1\}$-indicator function for the event $\mathcal{B}$; $h^{(\tau)}$ is a decreasing function of sweep τ. Note that $f(0,\tau) \equiv 1$ is the maximum.
- Catch rate vigilance r_0.

Step 2) (feed city and increment time t)

A city $\boldsymbol{x}$ is selected at random from $\boldsymbol{X}$. Find a winner $\boldsymbol{w}_m^{(t)}$ satisfying

$$\min_{0 \le m < M} \|\boldsymbol{x} - \boldsymbol{w}_m^{(t)}\|^2.$$

Step 3) (weight update and self-organization)

For $\boldsymbol{w}_m^{(t)}$ with $|m - \ell| \le h^{(\tau)}$, compute the following:

$$\begin{aligned}\boldsymbol{w}_m^{(t+1)} &= \boldsymbol{w}_m^{(t)} + \varepsilon^{(\tau)} f(m-\ell,\tau) I(|m-\ell| < h^{(\tau)})(\boldsymbol{x} - \boldsymbol{w}_m^{(t)}) \\ &\quad + \alpha^{(\tau)}(\boldsymbol{w}_{m+1}^{(t)} - 2\boldsymbol{w}_m^{(t)} + \boldsymbol{w}_{m-1}^{(t)}).\end{aligned} \tag{35}$$

Here, $m = \ell$ is the case of the winner update. The rest of the cases $m \neq \ell$ are for the self-organization.

Step 4) (test and termination)

The following test and update are executed at every sweep. If each city has a distinct winner, then the learning is completed. Otherwise, if the catch percentage is greater than r_0 and not increasing, there is a winner for multiple cities. This weight is copied. Then, $\varepsilon^{(\tau)}, \alpha^{(\tau)}, f(m,\tau)$ are updated according to (26) and one of (30)–(32). Then, go to Step 2).

2) Experiments: The USA 532 set [20] is used for testing the ability of the harmonic competition with dynamic control of the parameters. This data set contains $N = 532$ cities in the USA which are located quite nonuniformly. The initial neural weights are placed on an ellipse of the USA territory. The total number of neurons is $M = 2500$. Common specifications throughout the experiments are: $\varepsilon^{(0)} = 0.25$; $f(k,\tau) = \exp(-k^2/2\sigma^2(\tau))$; $\sigma(\tau) = \sigma(0)(1-s)^\tau$; and $h^{(\tau)} = 2\sigma(\tau)$. Fig. 10(a), (b), and (c) show the process of learning and the resulting tour. The adjustment rule for $\lambda^{(\tau)}$ uses (31) with $a = 2N^2 \times 10^{-7}$. Fig. 10(a) shows the initial configuration of neurons ($\tau = 0$). Fig. 10(b) illustrates the intermediate state of learning ($\tau = 85$). Fig. 10(c) is the resulting tour with a length of 8.9731 at $\tau = 130$. Fig. 10(d) illustrates the progress of learning by showing the parameter $\varepsilon^{(\tau)}$. One observes from this figure that the learning parameter $\varepsilon^{(\tau)}$ has an increasing trend. This is the dual case of Chapter III where $\varepsilon^{(\tau)}$ had a decreasing trend [Fig. 6(d)].

Table VII compares the performances of the previously obtained results of the static control [17] and the newly obtained ones by dynamic control. Note that there are three dynamic rules on the adjustment of $\lambda^{(\tau)}$; (30)–(32). Comparing the performance and the required computation, the dynamic rule of (31) is recommended. We note here that the method of Angéniol *et al.* [1] is omitted. This is because that method corresponds to the case of $\lambda^{(\tau)} \equiv 0$ which gives inferior results.

B. Extended Vehicle Routing Problems

The vehicle routing problem [4] arose from the multiple person TSP. In the vehicle routing problem, however, drivers visit cities and collect or deliver their items (demands). There is a maximum load for the acceptable amount for each vehicle. Thus, the target of optimization is both the route length and vehicle load.

The extended vehicle routing problems (EVRP's) tried here have more constraints to be satisfied: Each city has its own preference for specific vehicles. Rejected vehicles cannot visit the cities which refuse them. This property, together with the optimization of the tour and demands, is a good example showing multiplicative constraints.

1) Formulation and Algorithm: There are K vehicles which start from, and come back to, the same depot. The kth vehicle's subtour length is

$$D_k = \sum_{i_k = 0 \bmod N_k}^{N_k - 1} d(\boldsymbol{x}_{\pi(i_k)}, \boldsymbol{x}_{\pi(i_k+1)}), \quad (k = 0, \cdots, K-1).$$

$$\Delta\mu^{(\tau)} = \begin{cases} 0, & \text{if } \bar{f}^{(\tau)} \le \bar{f}^{(\tau-1)} \text{ and } \bar{g}^{(\tau)} \le \bar{g}^{(\tau-1)}, \\ \gamma_\mu^{\text{down}}(\bar{\eta}^{(\tau)} - \bar{\eta}^{(\tau-1)})/\bar{\eta}^{(\tau-1)}, & \text{if } \bar{f}^{(\tau)} \le \bar{f}^{(\tau-1)} \text{ and } \bar{g}^{(\tau)} > \bar{g}^{(\tau-1)}, \\ \gamma_\mu^{\text{up}}(\bar{\eta}^{(\tau)} - \bar{\eta}^{(\tau-1)})/\bar{\eta}^{(\tau-1)}, & \text{if } \bar{f}^{(\tau)} \ge \bar{f}^{(\tau-1)} \text{ and } \bar{g}^{(\tau)} \le \bar{g}^{(\tau-1)}, \\ \gamma_\mu^{\text{up}}|\bar{\eta}^{(\tau)} - \bar{\eta}^{(\tau-1)}|/\bar{\eta}^{(\tau-1)}, & \text{if } \bar{f}^{(\tau)} \ge \bar{f}^{(\tau-1)} \text{ and } \bar{g}^{(\tau)} > \bar{g}^{(\tau-1)}. \end{cases} \tag{33}$$

TABLE VII
TOUR LENGTHS OBTAINED FOR THE USA-532 SET

Method	Tour length	Computation
λ dynamically controlled by $\mu = a\tau/\varepsilon$ (31)	8.9731 (2.49% longer)	4.1 h (SS1)
λ dynamically controlled by autonomous rule of (32)	8.9868 (2.65% longer)	4.2 h (SS1)
λ dynamically controlled by $\mu = a\tau$ (30)	9.0628 (3.52% longer)	4.1 h
Static control of λ	9.0357 (3.21% longer)	2.8h (SS1)
Simulated annealing	9.1728 (4.44% longer)	0.33 h (SS1)
Elastic net	11.7521 (34.23% longer)	170 h
Branch and cut	8.7550 (0.00% longer)	6.0 h (Cyber 205)

1) Tour lengths of static control on λ, simulated annealing [21], and elastic net [6] were obtained from the study [17].
2) The tour length of "branch and cut" was given in [20]. This solution required the linear programming package XMP on a Cyber 205 supercomputer.
3) SS1 is a conventional workstation with a speed of 12.5 MIPS and 1.4 MFLOPS.
4) Simulated annealing [21] required many repeated trials to find the best cooling schedule.

Note that $\boldsymbol{x}_0$ is the depot. Every city is visited by a single vehicle. The total number of cities is $\sum_{k=0}^{K-1} N_k = N$. The total demand of the kth vehicle is

$$B_k = \sum_{i_k=0}^{N_k-1} b(\boldsymbol{x}_{\pi(i_k)}), \quad (k = 0, \cdots, K-1).$$

The grand total demand is $B = \sum_{k=0}^{K-1} B_k$. Thus, the basic vehicle routing problem is an optimization of $(D_0, \cdots, D_{K-1}; B_0, \cdots, B_{K-1})$ given a set of city positions and demands, $(\boldsymbol{x}_n, b(\boldsymbol{x}_n)), (n = 0, \cdots, N-1)$. In the extended vehicle routing problems, each city has a type for specifying acceptable vehicles, $\{q_n\}_{n=0}^{N-1}$. Here, q_n is an element of the power set of K vehicles excluding the null set. For instance, the city $\boldsymbol{x}_2$ accepts only even-numbered vehicles ($k = 0, 2, \cdots, \lfloor K/2 \rfloor$). Thus, the extended vehicle routing problem is a multiple criteria optimization problem of $(\boldsymbol{x}_n, b(\boldsymbol{x}_n), q_n), (n = 0, \cdots, N-1)$. We will discuss the following four types of problems:

[EVRP1] Minimize $\sum_{k=0}^{K-1} D_k$ as long as the city preference and the upper bound $B_k \leq B_k^*$ is met.

[EVRP2] Try EVRP1 with the further restriction that the $\max_k D_k$ is kept small.

[EVRP3] Try EVRP1 with the further restriction that the $\max_k B_k$ is kept small.

[EVRP4] Try EVRP1 with the further restriction that the $\max_k D_k$ and $\max_k B_k$ are jointly kept small.

The constraints imposed on the EVRP's are realized by the following multiplicative handicaps for harmonic competition

$$h_1(\boldsymbol{x}_n, \boldsymbol{w}_m, q_n) = \begin{cases} 1, & \text{if the city } \boldsymbol{x} \text{ accepts the vehicle } k, \\ \infty, & \text{otherwise.} \end{cases}$$

The infinite penalty is equivalent to "using other vehicles only." To suppress the maximum of the subtour lengths

$$h_2(\boldsymbol{x}_n, \boldsymbol{w}_m, q_n) = \frac{D_k}{D} \times \frac{D_k}{D - D_k}, \quad (k = 0, \cdots, K-1) \tag{36}$$

is further multiplied for competition cost. Here, $D = \sum_{k=0}^{K-1} D_k$. To enhance this penalty, the maximum of the subtour lengths D_k, say $D_{\max}$, is replaced by $D_{\max}(KD_{\max}/D)$. This is valid for $2D_{\max} < D$. In all of the experiments, this enhancement was used. For the minimization of the maximum of subtotal demands, the above handicap (36) with the substitutions of D_k by B_k and D by B are used. This handicap is denoted by h_3. Thus, for EVRP1 to EVRP4, the multiplicative handicaps h_1, h_1h_2, h_1h_3, and $h_1h_2h_3$ are used, respectively.

The dynamic control is the same as (31). The algorithm is as follows:

Algorithm for EVRP's:

Step 1) (initialization; $t = 0$, $\tau = \lfloor t/N \rfloor = 0$)

The following data set and initial values are given:

- Set of cities $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{N-1}\}$, $\boldsymbol{x}_n \in R^2$.
- City $\boldsymbol{x}_0$ is the depot.
- K vehicles with specified capacities.
- Each city has a fixed amount of demand $\{b_0, \cdots, b_{N-1}\}$.
- Each city specifies acceptable vehicles by $\{q_0, \cdots, q_{N-1}\}$.
- Neural weight vectors are located on K closed curves. K overlapping circles are allowed if tie-breakers of competition are incorporated. The K neural rings are as follows:

$$W_k^{(t)} = \{\boldsymbol{w}_{0,k}^{(t)}, \cdots, \boldsymbol{w}_{m,k}^{(t)}, \cdots, \boldsymbol{w}_{M_k-1,k}^{(t)}\}, \boldsymbol{w}_m^{(t)} \in R^2, \quad (k = 0, \cdots, K-1).$$

- Dynamic control rule of $\varepsilon^{(\tau)}$ and $\lambda^{(\tau)}$.
- Neuron's cooperation weight $f(m, \tau)I(|m| \leq h^{(\tau)})$.
- Initial handicap of each neuron is unity.
- Catch rate vigilance r_0.

Step 2) (feed city)

A city $\boldsymbol{x}$ is selected at random from the city set. If $\boldsymbol{x}$ is a regular city, go to Step 3). If it is the depot, go to Step 4).

Step 3) (update weights for a regular city).

Find a winner $\boldsymbol{w}_{\ell,k}^{(t)}$ such that

$$\boldsymbol{w}_{\ell,k}^{(t)} = \min_{0 \leq \kappa < K-1} \min_{0 \leq m < M_\kappa - 1} [\text{handicap}] \, \|\boldsymbol{x} - \boldsymbol{w}_{m,\kappa}^{(t)}\|^2.$$

For $\boldsymbol{w}_{\ell,k}^{(t)}$ itself and $\boldsymbol{w}_{m,k}^{(t)}$ with $|m - \ell| \leq h^{(\tau)}$, the weight vectors are updated as follows:

$$\begin{aligned} \boldsymbol{w}_{m,k}^{(t+1)} &= \boldsymbol{w}_{m,k}^{(t)} \\ &\quad + \varepsilon^{(\tau)} f(m-\ell, \tau) I(|m-\ell| \leq h^{(\tau)})(\boldsymbol{x} - \boldsymbol{w}_{m,k}^{(t)}) \\ &\quad + \alpha^{(\tau)}(\boldsymbol{w}_{m,k+1}^{(t)} - 2\boldsymbol{w}_{m,k}^{(t)} + \boldsymbol{w}_{m,k-1}^{(t)}). \end{aligned} \tag{37}$$

Here, $\alpha^{(\tau)} = \varepsilon^{(\tau)}\lambda(\tau)$. Then, go to Step 5).

Step 4)

For the depot $\boldsymbol{x} = \boldsymbol{x}_0$, find k winners $\boldsymbol{w}_{m,k}^{(t)}$ satisfying

$$\min_{0 \leq m < M_k - 1} \|\boldsymbol{x} - \boldsymbol{w}_{m,k}^{(t)}\|^2 \quad (k = 0, \cdots, K-1).$$

The update of the weights are the same as in (37). Then, go to Step 5).

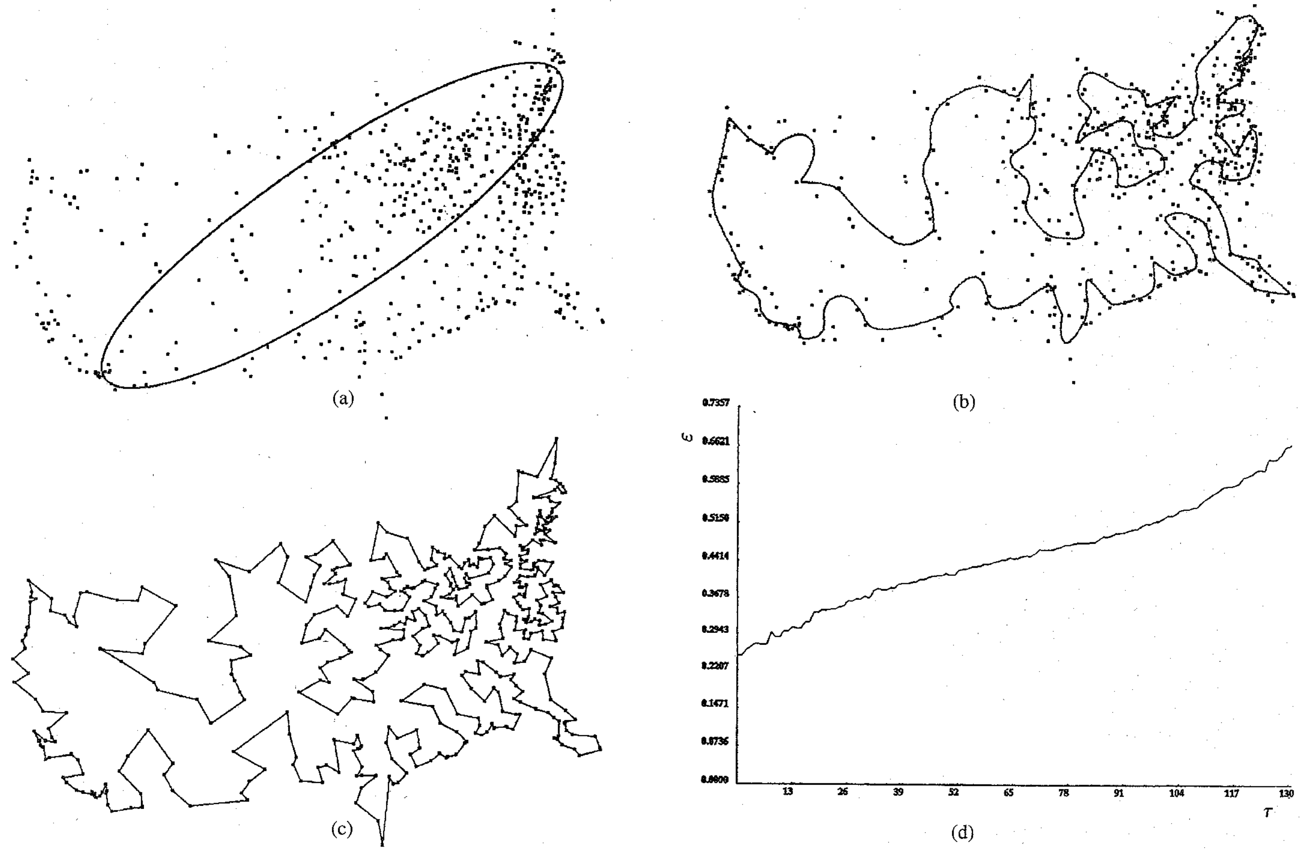


Fig. 10. Obtained tour and the progress of learning by the dynamic control of λ (length=8.9731; 2.49% longer than the true optimal). (a) Initial configuration of neurons ($\tau = 0$). (b) Progress of self-organization ($\tau = 85$). (c) Resulting tour ($\tau = 130$). (d) Dynamic control of ε obtained.

Step 5) (halt iteration and parameter update)

If there is a distinct winner for each city, the learning is completed. Else if the catch percentage is greater than r_0 and not increasing with the iterations, then there is a multiple winning neuron. The weight of such a neuron is copied. Parameters and functions such as $\varepsilon^{(\tau)}$ and $\alpha^{(\tau)}$ are then updated.

2) Experiments: In the following experiments, problems with four vehicles, $K = 4$, are selected. The data set is the USA-532 set. The depot $\boldsymbol{x}_0$ is $(0.6745, 0.6781)$. Each city's demand and city type are generated by the following method: If each city position is $(0.x_1x_2x_3x_4, 0.y_1y_2y_3y_4)$ in decimal floating point numbers, the demand is set to be $x_3 + y_3 (\mathrm{mod}\, 10)$ and the type number is $x_4 + y_4 (\mathrm{mod}\, 4)$. Type 0 cities (marked by "○" in the illustrations) accept only vehicle $k = 0$. Type 1 cities (marked by "△" in the illustrations) accept vehicles $k = 0$ and 1. Types 2 and 3 cities (marked by "□" in the illustrations) are treated as the same class: They accept any type of vehicles ($k = 0, 1, 2, 3$). Fig. 11(a) shows the initial state of neural weight vectors on four overlapping circles. Each circle contains $N = 532$ neurons. Thus, the total number of neurons is $M = 4N = 2128$ at the start. The city described by a filled square is the depot. Fig. 11(b) is the progress of the self-organization ($\tau = \lfloor t/N \rfloor = 1$). Fig. 11(c)–(f) are the results of EVRP1 to EVRP4. In these experiments, $a = N^2 \times 10^{-7}$. Other parameters are the same as those specified in Section IV. Table VIII compares the results of the presented dynamic method and the previous static case [17]. Underlined are numbers to be compared. For the EVRP1, only the total length is tested. In the case of the EVRP2, the total length and the maximum subtour length are evaluated. For the EVRP3, the total length and the maximum demand are checked. In the EVRP4, the total length and both maxima of the subtour length and the demand are compared. By this Table VIII, one finds that the results of the EVRP1, EVRP3 and EVRP4 are superior to those of the static method. Especially, in the EVRP4 which is the most difficult problem with optimization conflicts, the obtained answer is more desirable by far: It has a much shorter total length. The maximum subtour length is also shorter. The maximum demand is smaller too. Thus, we can conclude that the dynamic control presented in this paper is very effective.

V. Concluding Remarks

General competitive learning was discussed. The cost to be minimized included the main cost for data approximation and the subcosts for constraints reflecting the problem to be solved. The harmonic competition is a learning strategy to minimize such composite costs for multiple criteria optimization.

In this paper, we presented two classes of problems: multiple criteria vector quantization and traveling salesperson problems with their sophisticated versions. All of these are based on competitive learning with subcosts. Such a subcost approach can also be effective in problem solving with other learning

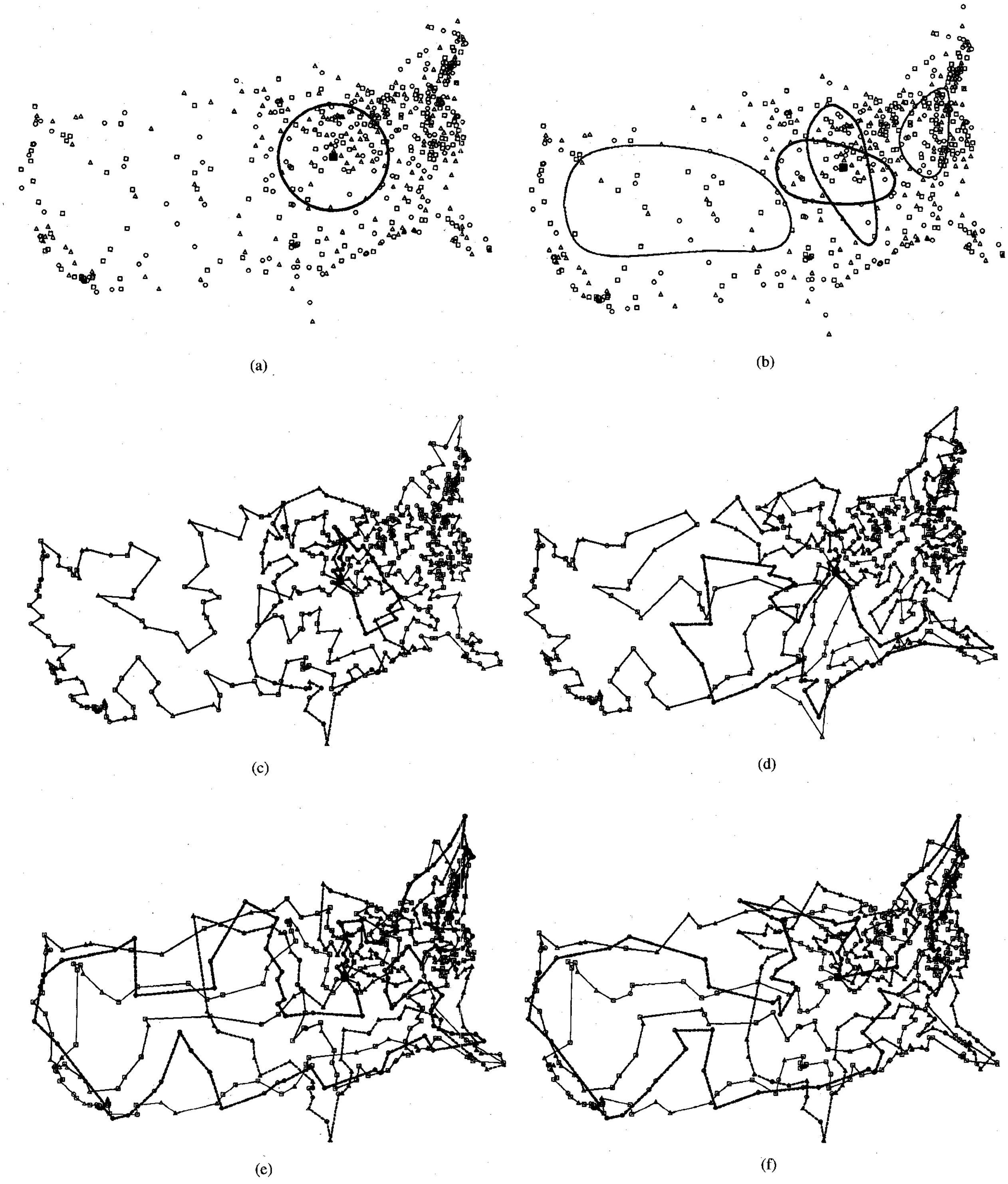


Fig. 11. Extended vehicle routing problems. (a) Initial configuration of neurons (four overlapping circles). (b) Progress of self-organization ($\tau = 1$). (c) Resulting tours of vehicles for EVRP1. (d) Resulting tours of vehicles for EVRP2. (e) Resulting tours of vehicles for EVRP3. (f) Resulting tours of vehicles for EVRP4.

paradigms. Such an extended study including the supervised learning is given in [18].

All algorithms are expected to minimize the total cost (4) every time a data $\boldsymbol{x}_n$ is fed. Finding a winner to learn was interpreted as minimal learning. This concept is not because of the minimization (5). Rather, such a notion came from the interpretation that the learning should be performed by the most appropriate neuron so that the system modification

TABLE VIII
COMPARISON OF THE DYNAMIC AND STATIC CONTROLS OF λ FOR THE EXTENDED VEHICLE ROUTING PROBLEMS

Strategy	Dynamic control			Static control		
	Total length	Sublength	Subdemand	Total length	Sublength	Subdemand
EVRP1	10.0105	0.5784	108	10.2421	0.8385	118
		1.7310	313		2.0294	442
		4.1710	768		4.4200	816
		3.5302	1188		2.9542	1001
EVRP2	11.5512	1.9980	231	11.4987	2.2389	304
		2.9379	737		2.8597	612
		2.9741	501		3.3520	863
		3.6412	908		3.0481	598
EVRP3	13.5380	3.8359	485	13.7066	3.6453	465
		3.2863	574		3.1572	595
		3.4034	654		3.4288	649
		3.0124	664		3.4753	668
EVRP4	13.0451	3.4628	410	13.3596	3.5326	415
		3.1641	628		3.1099	629
		3.3543	662		3.1979	680
		3.0639	677		3.5193	653

1) "Static control" were the results obtained in [17].

is kept minimal. Even if the problem is described as a maximization, the neuron of "arg max" instead of "arg min" is the minimal learning element. Suppose that nonminimal learning neurons were selected to learn. Such a learning strategy will not properly converge. Thus, the minimal learning is stable and effective by finding the most appropriate element to be modified. This interpretation remains valid for most of supervised learning: Only the minimum necessary modifications are applied to achieve the learning.

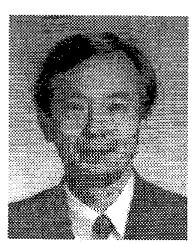
All of the aforementioned learning strategies were described based upon the Euclidian metric. The presented methods, however, remain valid for a class of non-Euclidian distortion measures [14]. Especially, the log-AEP can be derived for various distortion measures.

ACKNOWLEDGMENT

The author is grateful to A. Matsuno, T-S. Moon, K. Morisawa, M. Kobayashi, H. Nakayama, T. Sasai, Y-P. Chen, and T. Furuya for their discussions and software supports.

REFERENCES

[1] B. Angéniol, G. de La Croix Vaubis, and J.-Y. Le Texier, "Self-organizing feature maps and traveling salesman problem," *Neural Networks*, vol. 4, pp. 289–293, 1988.
[2] S. Amari and H. Nagaoka, *Methods of Information Geometry*. Tokyo: Iwanami, 1993 (In Japanese).
[3] J. Buhmann and H. Kühnel, "Vector quantization with complexity costs," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1133–1145, 1993.
[4] N. Christofides, "Vehicle routing," in *The Traveling Salesman Problem*, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, Eds. Chichester, U.K.: Wiley, pp. 431–488, 1985.
[5] D. DeSieno, "Adding a conscience to competitive learning," in *Proc. Int. Conf. Neural Networks*, vol. I, San Diego, CA, 1988, pp. 117–124.
[6] R. Durbin and D. Wilshaw, "An analog approach to the travelling salesman problem using an elastic net method," *Nature*, vol. 326, pp. 689–691, 1987.
[7] M. R. Geary and D. S. Johnson, *Computers and Intractability*. New York: Freeman, 1979.
[8] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 373–380, 1979.
[9] M. Ishikawa, "A structural learning of neural networks based on entropy criterion," in *Proc. Int. Joint Conf. Neural Networks*, Beijing, China, vol. II, 1992, pp. 375–380.
[10] J. N. Kapur and H. K. Kesavan, *Entropy Optimization Principles with Applications*. San Diego, CA: Academic, 1992.
[11] J. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Comm.*, vol. COM-28, pp. 84–95, 1980.
[12] J. Marschak and R. Radner, *Economic Theory of Teams*. New Haven, CT: Yale Univ. Press, 1972.
[13] Y. Matsuyama, "A note on stochastic modeling of shunting inhibition," *Biol. Cybern.*, vol. 24, pp. 139–145, 1976.
[14] ——, "Mismatch robustness of linear prediction and its relationship to coding," *Inform. Contr. (Inform. Computa.)*, vol. 47, pp. 237–262, 1980.
[15] ——, "Self-organization via competition cooperation and categorization applied to extended vehicle routing problems," in *Proc. Int Joint Conf. Neural Networks*, vol. I, Seattle, WA, 1991, pp. 385–390.
[16] Y. Matsuyama and M. Kobayashi, "Minimum learning with autonomous cost adjustment," in *Proc. Int. Joint. Conf. Neural Networks*, vol. II, Beijing, China, 1992, pp. 326–334.
[17] Y. Matsuyama, "Competitive learning among massively parallel agents: Applications to traveling salesperson problems," *Neural, Parallel, Scientif. Computa.*, vol. 1, pp. 181–198, 1993.
[18] ——, "Learning algorithms associated with penalties and human intelligence: From performance improvement to animation coding," in *Proc. Int. Conf. Artificial Neural Networks*, Tainan, Taiwan, 1994, pp. 547–560.
[19] J. E. Moody, "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems," in *Proc. Int. Joint Conf. Neural Networks*, vol. II, Beijing, China, 1992, pp. 841–846.
[20] M. Padberg and G. Rinaldi, "Optimization of a 532-city symmetric traveling salesman problem by branch and cut," *Opr. Res. Lett.*, vol. 6, pp. 1–7, 1987.
[21] W. H. Press, B. P. Flannery, S. A. Teukolski, and W. T. Vettering, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Univ. Press, 1988.
[22] N. Ueda and R. Nakano, "Competitive and selective learning method for designing optimal vector quantizers," in *Proc. Int. Conf. Neural Networks*, vol. III, San Francisco, CA, 1993, pp. 1444–1450.
[23] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman, "Generalization by weight-elimination applied to currency exchange rate prediction," in *Proc. Int. Joint Conf. Neural Networks*, Singapore, vol. 3, 1991, pp. 2374–2379.
[24] Y. Yamada, S. Tazaki, and R. M. Gray, "Asymptotic performance of block quantizers with a difference distortion measure," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 6–14, 1980.
[25] T. Yoshihara and T. Wada, "Optimization by extended VQ," in *Proc. Int. Joint Conf. Neural Networks*, vol. I, Seattle, WA, 1991, pp. 407–414.



Yasuo Matsuyama (S'77-M'78-SM'92) received the B. Eng., M. Eng. and Dr. Eng. degrees in 1969, 1971, and 1974, all from Waseda University, Japan. In addition, he received the Ph.D. degree from Stanford University, California, in 1978.

He worked at Ibaraki University as a Professor and Division Chairperson of the doctor course. In 1994, he was with the National Personnel Authority as a Cochairperson of the governmental personnel selection. Since 1996, he has been a Professor at Waseda University. His current research interests include computational and communication mechanisms, symbol/subsymbol learning algorithms, multimedia processing, and their electronic implementations.

Dr. Matsuyama received awards from the Institute of Electronics, Information, and Communication Engineers (IEICE) and the Electrical Communication Foundation. He has been a Councilor of the IEICE Tokyo Chapter since 1995.