Relaxation Labeling Using Augmented Lagrange-Hopfield Method

Stan Z. Li William Y. C. Soh Eam Khwang Teoh

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798

A novel relaxation labeling (RL) method is presented based on Augmented Lagrangian multipliers and the graded Hopfield neural network (ALH). In this method, an RL problem is converted into a constrained optimization problem and solved by using the augmented Lagrangian and Hopfield techniques. The ALH method yields results comparable to the best of the existing RL algorithms in terms of the optimized objective values, yet it is more suitable for analog neural implementation. Experimental results are presented.

Key words:
Augmented Lagrange method, constrained optimization, graded Hopfield networks, relaxation labeling.

1 Introduction

Relaxation labeling (RL), originally proposed by Rosenfeld *et al.* [1], is a class of parallel iterative numerical procedures which make use of contextual constraints to reduce ambiguities in image analysis. It is widely used to solve image processing and vision problems where contextual constraints play an important role. Such problems include image smoothing, enhancement, remote sensing, edge and boundary segmentation, motion analysis, perceptual organization, object matching and recognition, and so on.

Ullman [2] points out that mathematically, RL can be posed as a problem of constrained optimization. Faugeras and Berthod [3] define a class of global criteria in terms of transition probabilities. Haralick [4] illustrates RL as minimizing the expected loss from a Bayes viewpoint. Hummel and Zucker [5]

[1] Email address: szli@szli.eee.ntu.ac.sg

formulate RL as a global optimization problem in which a so called average local consistency measure is maximized.

A number of RL algorithms have been proposed. Rosenfeld *et al.* [1] propose basic RL algorithms for updating over discrete and continuous labeling spaces. Peleg [6] proposes a product of combination rule for fast convergence. A method known as gradient projection is used for the constrained optimization in [3,5]. Prince [7] compares several RL algorithms and shows that the two algorithms in [3,5] yield the best quality solutions.

More RL algorithms have been proposed recently: The simplex method of [8] provides quick convergence for the continuous RL. The radial projection method of [9] simplifies the projection operation used in [5]. The iterative conditional modes (ICM) algorithm [10], a discrete algorithm that uses a simple greedy method, has very fast convergence rate. The fast convergence of these algorithms is usually gained at the expense of loss in the solution quality as measured by the optimized objective values.

The simulated annealing (SA) algorithm can find the global minimum with probability approaching one, theoretically, when the annealing schedule is low enough [11]. The disadvantages of the SA are high computational cost, low convergence rate and difficulties in tuning the annealing schedule [12].

Apart from the solution quality and the convergence rate, another consideration in the evaluation of RL algorithms may be added, that is, the suitability for parallel and distributed, and even analog network, implementation. Existing continuous RL [1,6,3,5,9,8] are complicated in this regard because they require some complicated operations such as the normalization of labeling assignments and the gradient projection.

This paper presents a novel relaxation labeling method called Augmented Lagrangian-Hopfield (ALH) method based on the Augmented Lagrangian multipliers and the graded Hopfield neural network. In the ALH method, RL is formulated as a problem of constrained real optimization. The augmented Lagrange multiplier method [13,14] is used for optimization with the constraints and the Hopfield method [15,16] for bridging the gap between discrete and continuous optimization.

The ALH needs no gradient projection nor other normalization operations in its updating equations in keeping the labeling constraints. Therefore, it is more amenable for a neural network implementation than the exiting RL algorithms. Experiments show that the ALH produces good quality solutions in terms of the optimized objective values at a reasonable number of iterations. A recent result shows that the ALH method significantly improves the Hopfield type networks in solving the traveling salesman problem [17]. The ALH has also been used for image restoration and segmentation [18].

The rest of the paper is organized as follows: Section 2 introduces the continuous RL Method. Section 3 poses RL as a constrained optimization problem and presents the ALH method for solving it. Section 4 discusses the constrained optimization methods in connection to RL. Section 5 gives a neural network structure for the ALH computation. Section 6 presents the experimental results.

2 Continuous Relaxation Labeling

Let $\mathcal{S} = \{1, \ldots, m\}$ be a set of m sites (termed objects in the traditional RL literature) to be labeled and $\mathcal{L} = \{0, 1, \ldots, M\}$ be a set of $(M+1)$ labels to be assigned to the sites. In continuous RL, the labeling state for each site $i \in \mathcal{S}$ is represented by an $(M+1)$ position vector

$$p_i = [p_i(I) \mid I \in \mathcal{L}] \qquad \forall i \in \mathcal{S} \tag{1}$$

subject to the following *feasibility* constraints

$$\begin{cases} \sum_{I \in \mathcal{L}} p_i(I) = 1 \ \forall i \\ p_i(I) \geq 0 \qquad\quad \forall i, I \end{cases} \tag{2}$$

The first of the above is the normalization operation. The real value $p_i(I) \in [0, 1]$ reflects the strength with which i is assigned label I. Each p_i lies in a hyperplane in the non-negative quadrant portion of the multi-dimensional real space $[0, 1]^{\mathcal{L}}$. The set $p = \{p_i \mid i \in \mathcal{S}\}$ is called a *labeling assignment*. The feasible space for labeling assignments is

$$\mathbb{P} = \{p \mid p_i(I) \geq 0, \sum_{I \in \mathcal{L}} p_i(I) = 1, i \in \mathcal{S}, I \in \mathcal{L}\} \tag{3}$$

However, the final solution p^* must be *unambiguous*, that is,

$$p_i^*(I) = 0 \quad \text{or} \quad 1 \qquad \forall i, I \tag{4}$$

where the value of 1 means that i is unambiguously labeled I. Therefore, the space for the final labeling assignments is

$$\mathbb{P}^* = \{p^* \mid p_i(I) \in \{0, 1\}, \sum_{I \in \mathcal{L}} p_i(I) = 1, i \in \mathcal{S}, I \in \mathcal{L}\} \tag{5}$$

$\mathbb{P}^*$ is the set of "corners" of $\mathbb{P}$. Non-corner points in the continuous space $\mathbb{P}$ provide routes to the corners, *i.e.* the points in $\mathbb{P}^*$.

The continuous RL can be regarded as an iterative process which minimizes the following energy function

$$E(p) = \sum_{i \in \mathcal{S}} \sum_{I \in \mathcal{L}} r_i(I)\ p_i(I) + \sum_{i \in \mathcal{S}} \sum_{I \in \mathcal{L}} \sum_{i' \in \mathcal{S}, i' \neq i} \sum_{I' \in \mathcal{L}} r_{i,i'}(I, I')\ p_i(I)\ p_{i'}(I') \tag{6}$$

where $r_i(I)$ and $r_{i,i'}(I, I')$ are unary and binary compatibility functions. [2] The minimization is over $\mathbb{P}$ and finally over $\mathbb{P}^*$ and hence is constrained. The constrained minimization is performed based on the gradient $q = \nabla E(p) = \{q_i(I) \mid \forall (i, I)\}$ where the gradient components are given by

$$q_i(I) = \frac{\partial E}{\partial p_i(I)} = r_i(I) + 2 \sum_{i' \in \mathcal{S}, i' \neq i} \sum_{I' \in \mathcal{L}} r_{i,i'}(I, I')\ p_{i'}(I') \tag{7}$$

for symmetric $r_{i,i'}(I, I')$. The initial assignment is set to $p^{(0)} \in \mathbb{P}$. The current $p^{(t)}$ is updated into a new state $p^{(t+1)}$ according to

$$p^{(t+1)} \leftarrow \Phi(p^{(t)}, q^{(t)}) \tag{8}$$

where Φ is an updating operator. RL algorithms differ from one another in the choice of Φ.

Existing iterative update schemes normally involve a normalization process or alike to meet the feasibility conditions. In [1], for example, Rosenfeld *et al.* use the following fixed point iteration (for maximization)

$$p_i^{(t+1)}(I) \leftarrow \frac{p_i^{(t)}(I)(1 + q_i^{(t)}(I))}{\sum_J p_i^{(t)}(J)(1 + q_i^{(t)}(J))} \tag{9}$$

In [3,5], a sophisticated operation called gradient projection is used.

The mean field annealing (MFA) algorithm [19] approximates the global minimum of $E(f)$ by evaluating the partition function at the global saddle point

[2] To be more precise, $r_i(I)$ and $r_{i,i'}(I, I')$ are *compatibility* functions when $E(p)$ is to be *maximized*; they represent *incompatibilities* when $E(p)$ is to be *minimized*. But in this paper, the term compatibility is used for both cases.

of the effective energy. The effective energy is derived in [19] as

$$E_{eff}(p,q) = E(p) + T\left[\sum_i \sum_I p_i(I) q_i(I) + \sum_i \ln \sum_I e^{q_i(I)}\right] \tag{10}$$

The saddle points are among the roots of the following mean field theory equations

$$\nabla_p E_{eff}(p,q) = 0, \qquad \nabla_q E_{eff}(p,q) = 0 \tag{11}$$

where ∇_x is the gradient with respect to x. Adding to the effective energy a term $-\frac{\beta}{2}\sum_i \sum_I p_i^2(I)$ has a constructive balancing effect in solving the mean field equations [19] and when β is high enough, the minima of $E(p) - \frac{\beta}{2}\sum_i \sum_I p_i^2(I)$ always lies at the vertices of the hypercube [20]. With the additional term, we have the following the mean field theory equations

$$q_i(I) = -\frac{1}{T}\left[\frac{\partial E(p)}{\partial p_i(I)} - \beta p_i(I)\right], \qquad p_i(I) = \frac{e^{q_i(I)}}{\sum_J e^{q_i(J)}} \tag{12}$$

Schemes for selecting β values are provided in [19,20], which require knowing the eigenvalues of the Hessian of $E(p)$. We set $\beta = 20$ by trial and error; a smaller value would not ensure the convergence for some cases. Our MFA results were obtained with a synchronous updating procedure.

3 The Augmented Lagrange-Hopfield Method

RL can be posed as minimizing an energy function $E(p)$ subject to the constraints in (2). This is formally described as the following constrained optimization problem:

$$\min_p \quad E(p) \tag{13}$$

$$\text{subject to} \quad C_i(p) = \sum_I p_i(I) - 1 = 0 \qquad \forall i \in \mathcal{S} \tag{14}$$

$$p_i(I) \geq 0 \qquad \forall i \in \mathcal{S}, \forall I \in \mathcal{L} \tag{15}$$

where $E(p)$ is the objective energy function. The final solution p^* is subject to (4).

Now we present the ALH method for solving the above constrained minimization problem. First, we introduce internal variables $u_i(I) \in (-\infty, +\infty)$ $(\forall i, I)$

and relate them to $p_i(I)$ via a sigmoid function

$$p_i(I) = \psi_T(u_i(I)) = \frac{1}{1 + \mathrm{e}^{-u_i(I)/T}} \tag{16}$$

controlled by a temperature parameter $T > 0$. With the introduction of the internal u variables, the energy function can be considered as $E(u) = E(p(u))$. This treatment confines $p_i(I)$ to the range $(0, 1)$ to impose the inequality constraints of (15). When $T \to 0^+$, $p_i(I)$ is forced to be 0 or 1 depending on whether $u_i(I)$ is negative or positive, thus imposing the unambiguity constraints of (4).

Next, we use the Lagrange method to impose the equality constraints of (14). Define the following Lagrange function

$$L(p, \gamma) = E(p) + \sum_k \gamma_k C_k(p) \tag{17}$$

where γ_k are called the Lagrange multipliers. It is a function of the $M \times m$ variables of p and the m variables of γ. For p^* to be a local minimum subject to the constraints, it is necessary that (p^*, γ^*) be a stationary point of the Lagrange function:

$$\begin{aligned} \nabla_p L(p^*, \gamma^*) &= 0 \\ \nabla_\gamma L(p^*, \gamma^*) &= 0 \end{aligned} \tag{18}$$

If (p^*, γ^*) is a saddle point for which

$$L(p^*, \gamma) \leq L(p^*, \gamma^*) \leq L(p, \gamma^*) \tag{19}$$

then p^* is a local minimum of $E(p)$ satisfying (14), *i.e.* a local solution to the constrained optimization problem [21]. The following dynamic equations

$$\frac{\mathrm{d}p_i(I)}{\mathrm{d}t} = -\frac{\partial L(p, \gamma)}{\partial p_i(I)} \tag{20}$$

$$\frac{\mathrm{d}\gamma_i}{\mathrm{d}t} = +\frac{\partial L(p, \gamma)}{\partial \gamma_i} \tag{21}$$

called the basic differential multiplier method in [22], can be used to find such a saddle point. It performs energy descent on p but ascent on γ. The convergence of this system is illustrated in [23].

The Lagrangian (17) can be augmented by adding penalty terms $[C_k(p)]^2$, giving an augmented Lagrange function [13,14]

$$L_\beta(p,\gamma) = E(p) + \sum_k \gamma_k C_k(p) + \frac{\beta}{2}\sum_k [C_k(p)]^2 \tag{22}$$

where $\beta > 0$ is a finite weighting factor. The introduction of the quadratic term $\beta \sum_k [C_k(p)]^2$ with $C_i(p) = 0$ $(\forall i)$ does not alter the location of the saddle point. The quadratic terms can effectively stabilize the system [13,14]. The dynamic equations for minimizing L_β are

$$\frac{\mathrm{d}p_i(I)}{\mathrm{d}t} = -\frac{\partial L_\beta(p,\gamma)}{\partial p_i(I)} = -\frac{\partial E(p)}{\partial p_i(I)} - \sum_k \gamma_k \frac{\partial C_k(p)}{\partial p_i(I)} - \beta \sum_k C_k(p)\frac{\partial C_k(p)}{\partial p_i(I)} \tag{23}$$

$$\frac{\mathrm{d}\gamma_i}{\mathrm{d}t} = +\frac{\partial L_\beta(p,\gamma)}{\partial \gamma_i} = +C_i(p) \tag{24}$$

where

$$\frac{\partial C_k(p)}{\partial p_i(I)} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

for $C_i(p)$ defined in (14). The system performs energy descent on p but ascent on γ. This corresponds to the modified differential multiplier method in [22]. Our experiments show that the penalty terms are necessary for damping oscillations to help the convergence of the numerical computation.

In the ALH method, the updating of label assignments is performed on u, rather than on p, variables. Equ.(23) is replaced by

$$\frac{\mathrm{d}u_i(I)}{\mathrm{d}t} = -\frac{\partial L_\beta(p(u),\gamma)}{\partial p_i(I)} = -\left[\frac{1}{m}q_i(I) + \gamma_i + \beta C_i(p)\right] \tag{26}$$

where $q_i(I) = \frac{\partial E}{\partial p_i(I)}$ is weighted by a factor of $\frac{1}{m}$ to achieve some invariance to m. Because $\frac{\partial p_i(I)}{\partial u_i(I)} = \frac{\mathrm{e}^{-u_i(I)/T}}{T(1+\mathrm{e}^{-u_i(I)/T})^2}$ is always positive, $\frac{\mathrm{d}p_i(I)}{\mathrm{d}t}$ has the same sign as $\frac{\mathrm{d}u_i(I)}{\mathrm{d}t}$.

A damping term $\zeta u_i(I)$, where $\zeta > 0$ is a constant, may be added to (26) as in the graded Hopfield neural networks [15] to force $p_i(I)$ to approach 0 or 1. This is equivalent to adding an energy term

$$H(p) = \sum_k \sum_K \int_0^{p_k(K)} \psi_T^{-1}(p)\mathrm{d}p \tag{27}$$

to $L_\beta(p(u),\gamma)$, where the inverse of the sigmoid function ψ_T is

$$\psi_T^{-1}(p_i(I)) = u_i(I) = T \ln \frac{p_i(I)}{1-p_i(I)} \tag{28}$$

The term $H(p)$ reaches the minimum value of zero only when all $p_i(I)$'s are either 0 or 1. This means minimizing this term with $T \to 0^+$ in effect leads to the satisfaction of (4). However, this term is not necessary in our model because the satisfaction of (14) and (15) under a small T for $p_i(I) = 1/[1+e^{-u_i(I)/T}]$ is sufficient for the convergence to (4). [3] The competitive mechanism in (2) will make the winner take all.

Numerically, the ALH algorithm consists of the following three steps corresponding to (26), (16) and (24):

$$\begin{aligned}
u_i^{(t+1)}(I) &\leftarrow u_i^{(t)}(I) - \mu\left\{\frac{1}{m}q_i^{(t)}(I) + \gamma_i^{(t)} + \beta C_i(p^{(t)})\right\} && (29)\\
p_i^{(t+1)}(I) &\leftarrow \psi_T(u_i^{(t+1)}(I)) && (30)\\
\gamma_i^{(t+1)} &\leftarrow \gamma_i^{(t)} + \beta C_i(p^{(t)}) && (31)
\end{aligned}$$

In this procedure, μ is a step size; and during the update, T may be decreased, and β, which is also used as a step size in (31), may be increased to speedup convergence. The updating is performed synchronously in parallel for all i and I.

4 Discussions on Penalty and Lagrange Methods

When $\gamma_i = 0$ for all i, $L_\beta(p,\gamma) = E(p) + \beta\sum_k [C_k(p)]^2$ and the augmented Lagrange method reduces to the penalty method. To approximate the constrained minima of $E(p)$ with $C_i(p) = 0$ by the unconstrained minima of $E(p) + \beta\sum_k [C_k(p)]^2$, it is necessary to send β to ∞. In numerical computation, however, minimizing $E(p) + \beta\sum_k [C_k(p)]^2$ becomes increasingly difficult as $\beta \to \infty$. This is due to the ill-conditioning of the problem for large β [24].

When $\beta = 0$, $L_\beta(p,\gamma)$ reduces to the standard Lagrange method of (17). In this simple form, the constraints cause the oscillation or zigzagging problem [24], which slows down the convergence or leads to divergence. Our experiments with systems without quadratic penalty terms, such as those presented in [2]

[3] T needs not be very low. We set $T = 10000$ in our experiments and obtained good convergence.

and [25] for solving the RL problem of the form (6), show that oscillations are always a problem with such systems.

The augmented Lagrange method, which combines both the Lagrange and the penalty methods, effectively overcomes the problems associated with the penalty method or the Lagrange method when used alone. From the viewpoint of duality theory, it is simply the standard Lagrange method for the constrained optimization with $E(p)$ in (13) replaced by $E(p) + \beta \sum_k [C_k(p)]^2$. The latter has the same constrained minima as the former when the constraints $C_i(p) = 0$ are satisfied. When $\beta > \beta_0$ where $\beta_0 > 0$ is some constant, the Hessian of $L_\beta(p, \gamma)$ will be positive definite at a saddle point (p^*, γ^*) [23]. This "convexifies" the problem and thus p^* is a local minimum of $E(p)$ with $C(p^*) = 0$ [23,26]. With the augmented Lagrangian, it is not necessary to send β to ∞ in order to induce the correct solution. In other words, the β value required by the augmented Lagrangian is much smaller than that required by the penalty method. Thus the ill-conditioning caused by large β is avoided. Furthermore, the convergence rate of the augmented Lagrange method is considerably better than the penalty method. A lower β value is helpful for achieving a better objective value.

Ullman [2] uses a result given in [23] to solve optimization problems constrained by (13) and (14). Without using the u variables as we do, the Lagrangian there is as in (17). The following update iterations is performed

$$p_i(I) \leftarrow \max\left[0, p_i(I) - \mu \frac{\partial L(p,\gamma)}{\partial p_i(I)}\right] \tag{32}$$

$$\gamma_i \leftarrow \max\left[0, \gamma_i + \mu \frac{\partial L(p,\gamma)}{\partial \gamma_i}\right] \tag{33}$$

Obviously, the inequality constraints (15) are maintained by the truncation operation implemented by the "max" operation while the equality constraints (14) are realized by using the multiplier. We believe the truncation in the second equation is not necessary; it is enforced as a heuristic that γ_i's are always positive for the considered RL problem. The system has been applied to motion correspondence where the objective is of the form [2]

$$E(p) = \sum_{i \in \mathcal{S}} \sum_{I \in \mathcal{L}} r_i(I)\ p_i(I) \tag{34}$$

The system converges to an unambiguous labeling, which satisfies the unambiguity constraints (4), for the above objective. However, it does not necessarily converge for the general RL problem in the form of (6). More critically, it suffers from divergence for the general RL problem. Quadratic penalty terms must be added for the convergence; in other words, the Lagrange function must be augmented by a penalty term (such as the quadratic penalty) for the system to converge.

Platt and Barr [22] apply the Arrow-Hurwicz differential equations to solve the traveling salesman problem formulated in terms of an elastic net [27]. Wacholder *et al.* Wacholder-89 combine Platt and Barr's result with Hopfield method [15] and give an energy which includes a term like (27). Because the stabilizing quadratic term is not used, it would also suffer from the previously mentioned zigzagging problem.

5 ALH Neural Network

Compared to the existing RL algorithms, the ALH is more amenable to neural network implementation. The Faugeras-Berthod [3] and Hummel-Zucker [5] algorithms involve an operation called "gradient projection", a complicated operation. Other RL algorithms need normalization operation to satisfy the first part of (2). It is not straightforward to implement these operations in a distributed way using analog neural network circuits. Such operations are not needed by the neuron-like ALH method.

To see how the ALH may be implemented using neural circuits, we substituting (7) and (14) into (26) and (24) to yield

$$\kappa_u \frac{\mathrm{d}u_i(I)}{\mathrm{d}t} = -\frac{1}{m}\left[r_i(I) + 2\sum_{i'}\sum_{I'} r_{i,i'}(I,I')\; p_{i'}(I')\right] - \gamma_i - \beta\left[\sum_{I'} p_i(I') - 1\right] \qquad (\forall i \in \mathcal{S}, I \in \mathcal{L}) \qquad (35)$$

$$\kappa_\gamma \frac{\mathrm{d}\gamma_i}{\mathrm{d}t} = \sum_{I'} p_i(I') - 1 \qquad (\forall i \in \mathcal{S}) \qquad (36)$$

where $p_i(I) = \psi_T(u_i(I))$, and κ_u and κ_γ are added to control the stability and convergence. Equations (35) can be rewritten as

$$\kappa_u \frac{\mathrm{d}u_i(I)}{\mathrm{d}t} = \sum_{I'}\sum_{i'} R_{i,i'}(I,I')\; p_{i'}(I') - \gamma_i + R_i(I) \qquad (\forall i \in \mathcal{S}, I \in \mathcal{L}) \qquad (37)$$

where $R_i(I) = -\frac{1}{m} r_i(I) + \beta$ and

$$R_{i,i'}(I,I') = \begin{cases} -\frac{2}{m} r_{i,i'}(I,I') - \beta & \text{if } i = i' \\ -\frac{2}{m} r_{i,i'}(I,I') & \text{otherwise} \end{cases} \qquad (38)$$

The ALH network implements Equations (37) and (36). The network consists of $m \times (M+1)$ neurons, indexed by (i, I) for $i \in \mathcal{S}, I \in \mathcal{L}$. The state of neuron (i, I) is described by the effective input potential $u_i(I)$ and the output value $p_i(I)$. The m variables of γ_i serve as hidden neurons.

This is basically the graded neural network of Hopfield style [16], extended by the hidden neurons. The strength of the synaptic current from a presynaptic neuron (i', I') to the post-synaptic neuron (i, I) is proportional to the product of $p_{i'}(I')$ (the presynaptic output) and $R_{i,i'}(I, I')$ (the strength of the synapse from (i', I') to (i, I)). Equations (37) express the net input current charging the input capacitance κ_u of neuron (i, I) as the sum of three sources: (i) postesynaptic currents induced in neuron (i, I) by presynaptic activity in neuron (i', I'), (ii) an input current γ_i due to the potential of the hidden neuron i (the synaptic strength from hidden neuron i to neuron (i, I) is the unity), and (iii) an input current due to the fixed external bias $R_i(I)$. The net input current charging the input capacitance κ_γ of hidden neuron i consists of two sources: (i) postesynaptic currents induced in hidden neuron i by presynaptic activity in neuron (i, I'), and (ii) a fixed input current -1. The two groups of neurons, $p_i(I)$ and γ_i, are coupled to each other for collective computation.

6 Experiments

In the following experiments, we compare (i) the ALH algorithm with the following three RL algorithms: (ii) that of Hummel and Zucker [5], the best existing RL algorithm according to a comparison [7]; (iii) the original RL of Rosenfeld *et al.* [1]; (iv) the mean field annealing (MFA) of Peterson and Soderberg [19]. Results obtained by using (v) the simulated annealing (SA) algorithm of [28], implemented based on a SA procedure given in [29], are also included as a reference. The comparison is in terms of (i) the solution quality measured by the optimized objective value, and (ii) the convergence rate measured by the number of required iterations.

6.1 Methods

The test bed is partial matching of weighted graphs in which the optimal node correspondences between the largest common parts of the two weighted graphs are to be established. Here, a node corresponds to a point in the $X-Y$ plane. Two random graphs, each containing $m = M$ nodes, are randomly generated as follows. First, a number of $\lfloor \frac{2}{3} m \rfloor$ nodes, where $\lfloor \cdot \rfloor$ is the "floor" operation, are generated using random numbers uniformly distributed within a box of size 100×100. They are for the first graph. Their counterparts in the other graph are generated by adding Gaussian noise of $N(0, \sigma^2)$ to the x and y coordinates of each of the nodes in the first graph. This gives $\lfloor \frac{2}{3}m \rfloor$ deviated locations of the nodes for the second graph. Then, the rest of the $\lceil \frac{1}{3}m \rceil$ nodes in each of the two graphs are generated independently using random numbers uniformly distributed in the box, which simulates outlier nodes. Between each pair of nodes in a graph is a weight equal to the Euclidean distance between them.

The above steps generate two weighted graphs $\mathcal{G} = (\mathcal{S}, d)$ and $\mathcal{G}' = (\mathcal{L}_0, D)$ where $\mathcal{S} = \{1, \ldots, m\}$ and $\mathcal{L}_0 = \{1, \ldots, M\}$ index the sets of nodes and $d = [d(i, i')]_{i,i' \in \mathcal{S}}$ and $D = [D(I, I')]_{I,I' \in \mathcal{L}_0}$ are the distances between nodes in the first and second graph, respectively. The basis of matching is the weights in d and D which reflect bilateral relations between nodes; unary properties of nodes are not used in the test. $\mathcal{L}$ is augmented to $\mathcal{L} = \{0, 1, \ldots, M\}$ by adding a special node, called the NULL node and indexed by 0, to cater to the matching of outlier nodes.

The matching is to assign a node from $\mathcal{L}$ to each of the nodes in $\mathcal{S}$ so that the energy E is minimized. The assignment of labels determines the node correspondences between the two graphs. The energy $E(f)$ sums up the costs of differences between the matched graph weights ($d(i, i')$ and $D(I, I')$ given $i \leftrightarrow I$ and $i' \leftrightarrow I'$) and the costs of the NULL matches (extra nodes in $\mathcal{S}$) under f. These costs are determined by the interaction matrix $[r_{i,i'}(I, I')]$.

The interaction matrix is defined in the following way

$$r_{i,i'}(I, I') = \begin{cases} [d(i, i') - D(I, I')]^2 & \text{if } I \neq 0 \ \&\ I' \neq 0 \\ r_{20} & \text{otherwise} \end{cases} \tag{39}$$

where $r_{20} > 0$ (a value of 0.7 is used in the experiments) is a constant. The compatibility $r_{i,i'}(I, I')$ measures the cost of the pair of matches $(i \leftarrow I)$ and $(i' \leftarrow I')$ when $I, I' \neq 0$. It takes the zero value when $d(i, i') = D(I, I')$, meaning that the pair of matches are perfect. It is greater than zero when $d(i, i') \neq D(I, I')$. When either of I and I' is 0 (NULL match), $r_{i,i'}(I, I')$ takes a prescribed value r_{20}.

To reduce storage, we used only one byte (8 bits) to represent the compatibility coefficients $r_{i,i'}(I, I')$. Positive values are truncated to an integer between 0 and 255 while negative values are truncated to zeros. This a low precision representation.

The above steps construct the compatibility matrix $[r_{i,i'}(I, I')]$. The ultimate objective of the comparison is more abstract: It is to compare the ability of the algorithms to solve the constrained minimization expressed in Equations 13–

15 and 4 in terms of the optimized objective value and convergence rate, given the compatibility matrix.

The parameters for the ALH equations (29) – (31) are set as follows: $\beta = 100$, T decreased from 100000 to 10000 according to $T^{(t+1)} \leftarrow 0.9T^{(t)}$, β increased from 1 to 100 according to $\beta \leftarrow 1.01\beta$, and $\mu = 1000$. The annealing schedule for the SA is $T^{(t+1)} \leftarrow 0.99T^{(t)}$ with the initial temperature set to $T^{(0)} = 10^9 \times m$. For the MFA, it is $T^{(t+1)} \leftarrow \kappa T^{(t)}$, where $\kappa = 0.95$, with $T^{(0)} = 10^9 \times m$ and $\beta = 20$; the κ and β values are chosen among $\kappa \in \{0.90, 0.95, 0.99\}$ and $\beta \in \{0, 5, 10, 20, 50, 100\}$ to produce the best results in terms of the solution quality.

Initial labeling is assigned as follows: First, we set $p_i^{(0)}(I) = 1/(M+1) + 0.001 * rnd$ as the starting point where rnd is a random number evenly distributed in $[-1, 1]$. This is the same for all the compared algorithms. For the ALH, the updated variables are initialized as $u_i^{(0)}(I) = \psi_T^{-1}(p_i^{(0)}(I))$ and $\gamma_i = 0$. For the SA algorithm, we set initial label for i as $I^* = \arg\max_I p_i^{(0)}(I)$.

The convergence of the continuous RL algorithms is judged by checking whether $\max_{i,I} |p_i^{(t+1)}(I) - p_i^{(t)}(I)|$ is smaller than $0.01/m^2$. For the SA, the algorithm terminates if the desired number of acceptances is not achieved at 100 successive temperatures [28]. For the MFA, the condition is the saturation value $S = \frac{1}{m} \sum_i \sum_I p_i^2(I) \geq 0.99$.

6.2 Results

Fig.1 compares the solution quality and convergence rate of the algorithms for the weighted graph matching problems of sizes $m = M = 10$ and 20, respectively, under various noise levels. Each value in these results is obtained by averaging 200 runs of which each run finds a minimum p^* given a distinct pair of graphs. The upper panel of the figure illustrates the solution quality in terms of the maximized gain defined as $G(p^*) = Const - E(p^*)$ where $Const$ is some constant. The higher the gain, the better the quality. When $m = 10$, the SA produces the best quality solutions, and the HZ, MFA, and ALH algorithms produce comparable results. The Rosenfeld solution deteriorates more quickly as the noise level increases. When $m = 20$, the SA, ALH and HZ perform almost equally well while the MFA and Rosenfeld solutions deteriorate more quickly as the noise level increases.

The lower panel of Fig.1 illustrates the convergence rate in terms of the required number of iteration before an algorithm converges. The lower, the better. The plots show that the continuous RL algorithms have about the same order of convergence rate, about hundreds of iterations. The SA is about 500 times slower. The figures demonstrate that the ALH produces solutions of quality comparable to the best of the RL algorithms at a reasonable number of iterations.

Fig.2 illustrates the evolution of the ALH system by showing the values of $p_i^{(t)}(I)$, $C_i^{(t)}(p)$ and $\gamma_i^{(t)}$ (for each i and I) as functions of the time t. The $p_i^{(t)}(I)$ variables (there are $m \times (M+1)$ of them) initially takes the values of $p_i^{(0)}(I) = 1/(M+1) + 0.001 * rnd$ at $t = 0$ (as stated earlier), and approach either 1 or 0 after hundreds of iterations, finally satisfying (4). In the meantime, the m C_i variables converge to 0 to satisfy (2). The m multipliers γ_i become stabilized after that. These manifest the convergence of the ALH algorithm. The noise level in this test is fixed as $\sigma = 2.0$.

7 Conclusion

With the use of Lagrange multipliers as in the augmented Lagrangian, the weighting values for the penalty terms can be much smaller than those required by the penalty method; thus the relative weight for the original objective is increased, which helps yielding a lower objective value. At the same time, the augmented Lagrange method also overcomes the zigzagging problem with the standard Lagrange method and improve the convergence. These advantages of the augmented Lagrangian plus the use of Hopfield method for imposing the inequality and unambiguity constraints have made the success of the ALH method. The experiments show that the ALH produces solutions of quality comparable to the best of the RL algorithms at reasonable computational expenses.

It is interesting to explore mathematical properties of the ALH algorithm. For example, under what conditions is the Hessian matrix for the ALH energy positive definite. How are they related to the involved parameters. These will lead to the further understanding of the ALH method.

Acknowledgement This work was supported by NTU Applied Research Project RG 43/95.

References

[1] A. Rosenfeld, R. Hummel, and S. Zucker. "Scene labeling by relaxation

operations". *IEEE Transactions on Systems, Man and Cybernetics*, 6:420–433, June 1976.

[2] S. Ullman. "Relaxation and constraint optimization by local process". *Computer Graphics and Image Processing*, 10:115–195, 1979.

[3] O. D. Faugeras and M. Berthod. "Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:412–423, April 1981.

[4] R. M. Haralick. "Decision making in context". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):417–428, July 1983.

[5] R. A. Hummel and S. W. Zucker. "On the foundations of relaxation labeling process". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3):267–286, May 1983.

[6] S. Peleg. "A new probability relaxation scheme". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:362–369, 1980.

[7] K. E. Price. "Relaxation matching techniques – A comparison". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5):617–623, September 1985.

[8] X. Zhuang, R.M. Haralick, and H. Joo. "A simplex-like algorithm for the relaxation labeling process". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1316–1321, 1989.

[9] P. Parent and S.W. Zucker. "Radial projection: an efficient update rule for relaxation labeling". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:886–889, 1989.

[10] J. Besag. "On the statistical analysis of dirty pictures" (with discussions). *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.

[11] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[12] E. H. L. Aarts. *Simulated annealing and Boltzmann machines : a stochastic approach to combinatorial optimization and neural computing.* Wiley, 1989.

[13] M. J. D. Powell. "A method of nonlinear constraints in minimization problems". In R. Fletcher, editor, *Optimization*, London, 1969. Academic Press.

[14] M. R. Hestenes. "Multipler and gradient methods". *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[15] J. J. Hopfield. "Neurons with graded response have collective computational properties like those of two state neurons". *Proceedings of National Academic Science, USA*, 81:3088–3092, 1984.

[16] J. J. Hopfield and D. W. Tank. " 'Neural' computation of decisions optimization problems". *Biological Cybernetics*, 52:141–152, 1985.

[17] S. Z. Li. "Improving convergence and solution quality of Hopfield-type neural network with augmented Lagrange multipliers". *IEEE Transactions on Neural Networks*, 7(6):1507–1516, November 1996.

[18] S. Z. Li, K. L. Chan, and H. Wang. "Bayesian image restoration and segmentation by constrained optimization". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–6, San Francisco, CA, June 16-20 1996.

[19] C. Peterson and B. Soderberg. "A new method for mapping optimization problems onto neural networks". *International Journal of Neural Systems*, 1(1):3–22, 1989.

[20] A. L. Yullie and J. J. Kosowsky. "Statistical physics algorithms that converge". *Neural Computation*, 6:341–356, 1994.

[21] B. S. Gottfried. *Introduction to Optimization Theory.* Prentice-Hall, 1973.

[22] J. C. Platt and A. H. Barr. "Constrained differential optimization". In *Proceedings of the IEEE 1987 NIPS conference*, 1988.

[23] K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Nonlinear Programming.* Stanford University Press, 1958.

[24] R. Fletcher. *Practical Methods of Optimization.* Wiley, 1987.

[25] E. Wacholder, J. Han, and R. C. Mann. "A neural network algorithm for the multiple traveling salesman problem". *Biological Cybernetics*, 61:11–19, 1989.

[26] Dimitri P. Bertsekas. *Constrained optimization and Lagrange multiplier methods.* Academic Press, New York, 1982.

[27] R. Durbin and D. Willshaw. "An analog approach to the travelling salesman problem using an elastic net method". *Nature*, 326:689–691, 1987.

[28] S. Kirkpatrick, C. D. Gellatt, and M. P. Vecchi. "Optimization by simulated annealing". *Science*, 220:671–680, 1983.

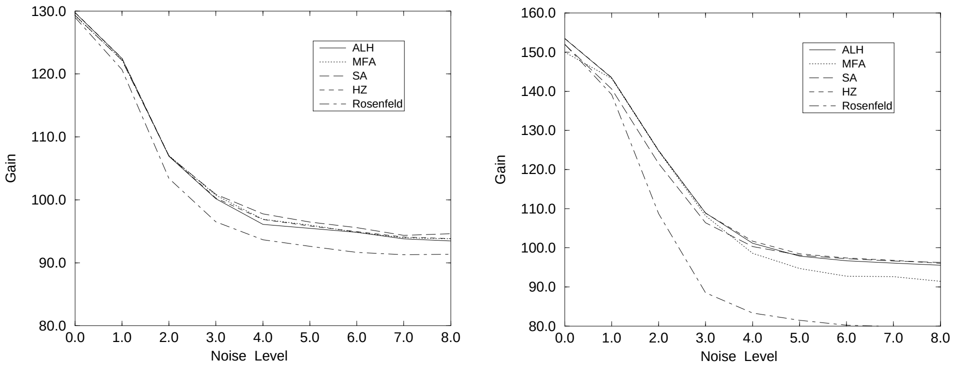
[29] William H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C.* Cambridge University Press, 2 edition, 1988.

About the Authors

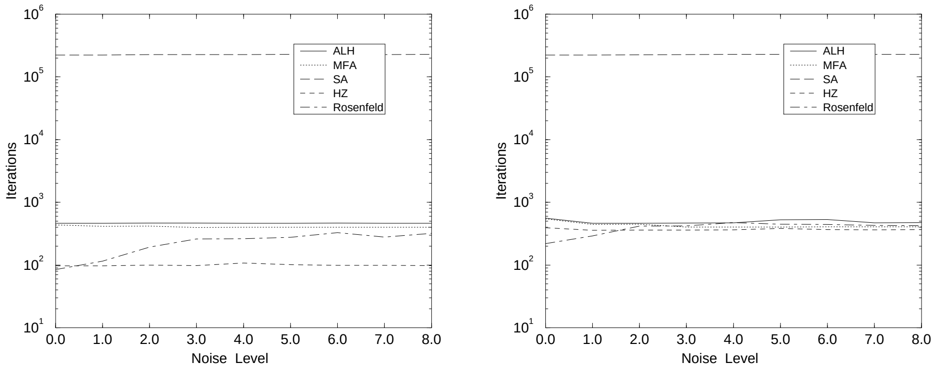
Stan Z. Li received the B.Eng degree from Hunan University, China, in 1982, M.Eng degree from the National University of Defense Technology, China, in 1985 and Ph.D degree from the University of Surrey, UK, in 1991. All degrees are in EEE. His research interests include computer vision, pattern recognition, image processing and optimization methods. He authored the book *Markov Random Field Modeling in Computer Vision* (Springer-Verlag, 1995).

William Y.C. Soh received his B.Eng in electrical and electronic engineering from the University of Canterbury, New Zealand, in 1983, and his Ph.D degree in electrical engineering from the University of Newcastle, Australia in 1987. From 1986 to 1987, he was a research assistant in the Department of Electrical and Computer Engineering, University of Newcastle. He joined the Nanyang Technological University, Singapore, in 1987 where he is currently a senior lecturer and the Head of the Control and Instrumentation Division, School of Electrical and Electronic Engineering. His current research interests are in the area of robust system theory, robotics, estimation and filtering, and signal processing. He has published over 70 research articles.

Eam Khwang Teoh received his B.E. and M.E. degrees in Electrical Engineering from the University of Auckland, New Zealand in 1980 and 1982 respectively and the Ph.D. degree in Electrical and Computer Engineering from the University of Newcastle, New South Wales in 1986. Presently, he is the Vice-Dean (Research) of the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is a co-inventor of an Australian Patent on An Adaptive Thickness Controller for a Rolling Mill. His current research interests are in the field of computer Vision and Pattern Recognition, Intelligent Systems, Robotics and Industrial Automation. He has published more than 120 journal and conference papers in these areas.

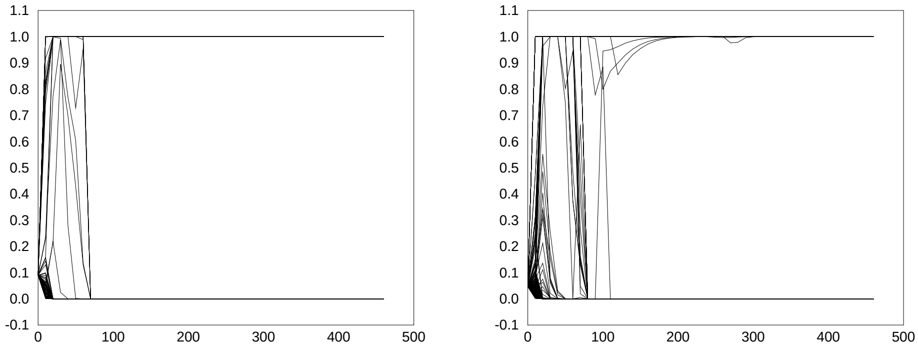


Maximized gain $G(p^*)$

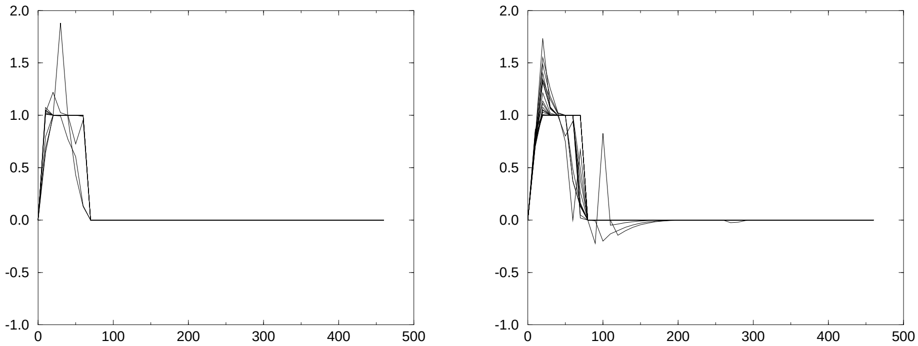


Number of iterations

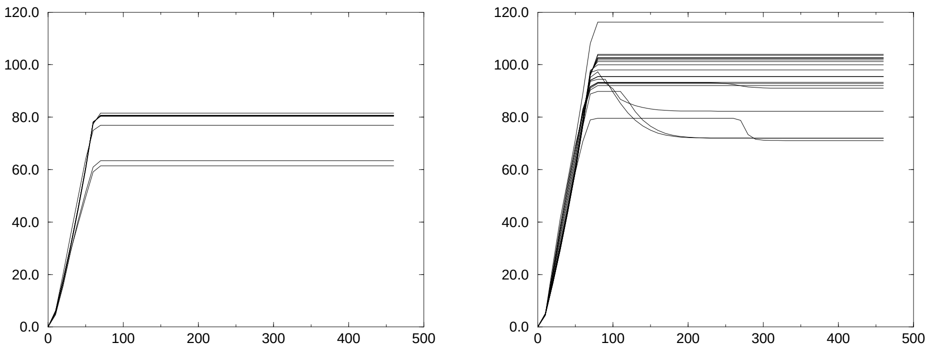
Fig. 1. The solution quality and the number of iterations needed, under various noise levels σ. Left: problem size $m = M = 10$. Right: $m = M = 20$.



Trajectories of $p_i^{(t)}(I)$ variables



Trajectories of $C_i^{(t)}(p)$ variables



Trajectories of $\gamma_i^{(t)}$ variables

Fig. 2. The evolution of the ALH system for problems of size $m = M = 10$ (left) and $m = M = 20$ (right). Horizontal is the time axis t.