Coupled Local Minimizers: Alternative Formulations and Extensions

J.A.K. Suykens, J. Vandewalle

Katholieke Universiteit Leuven
Department of Electrical Engineering ESAT-SISTA
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Tel: 32/16/32 18 02, Fax: 32/16/32 19 70
E-mail: johan.suykens@esat.kuleuven.ac.be

Abstract

Coupled local minimizers (CLM) is a new method for cooperative search which incorporates principles of state synchronization and cellular neural networks (CNN). Information between the optimizers is exchanged through coupling terms. The interaction between the individual minimizers can be optimized by solving linear programs. The original CLM method has been formulated within a Lagrange programming framework with equality constraints imposed between the individual optimizers. In this paper we discuss alternative formulations and extensions to the original formulation.

Keywords. Coupled local minimizers, cooperative search, synchronization, cellular neural networks.

1 Introduction

At whatever scale we look in the universe, life is full of interaction. Cellular Neural Networks (CNN) [3, 4] offers a methodology for describing interactions within a circuit-and systemtheoretical framework, either with respect to real-life engineering applications (such as image processing, robotics, cryptography etc.) or at a more abstract level of nonlinear dynamical systems and neural networks in general.

Recently, a method of Coupled Local Minimizers (CLM) has been proposed by Suykens *et al.* [9]. Within the CNN framework this CLM method can be viewed as an array that consists of cells of local minimizers that are locally coupled (Fig.1-2). Thanks to the coupling mechanism the optimizers are able to exchange information and perform better than a multistart local optimization method. The coupling terms result from pairwise state synchronization constraints imposed between the individual local minimizers. The coupling terms are related to an augmented Lagrangian [6] from which the penalty parameters can be optimized in such a way that an overall objective function (e.g. the average cost of the ensemble) can be minimized. This can be done by solving a simple linear program (LP). This approach implicitly incorporates principles of master-slave synchronization. Minimizers which are performing well can on-line impose their state to the others and the interaction can be optimized by solving the LPs. The CLM method has been successfully applied to problems of the optimization of Lennard-Jones clusters (related to the area of protein folding) and problems of learning and generalization of multilayer perceptrons [9].

In the original CLM formulation one optimizes the average cost of the ensemble subject to a set of pairwise state synchronization equality constraints. In this paper we discuss an alternative formulation which has a smaller number of inequality constraints, finally resulting in a smaller number of equations to be simulated. We also discuss how additional interaction terms can be included instead of pairwise coupling.

This paper is organized as follows. In Section 2 we discuss the original CLM formulation. In Section 3 we discuss cooperative CLM search and its link with master-slave synchronization. In Section 4 we present the alternative formulation. In Section 5 some remarks on full versus sparse interaction schemes are given. In Section 6 we give an illustrative example.

2 Original CLM formulation

Consider the minimization of a twice continuously differentiable cost function $U(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^n$. Let us take an ensemble that consists of a number of q local mini-

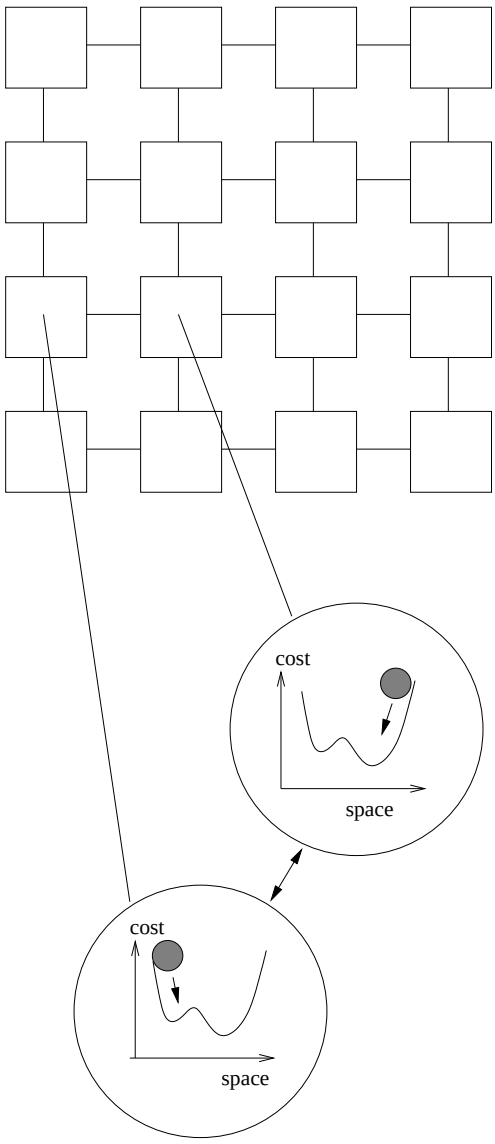


Figure 1: *CNN interpretation of CLMs.*

mizers. We aim at minimizing the average energy cost $\langle U \rangle = \frac{1}{q} \sum_{i=1}^{q} U[\mathbf{x}^{(i)}]$ of this ensemble subject to pairwise state synchronization of the particles [9]

$$\begin{array}{ll} \min\limits_{\mathbf{x}^{(i)} \in \mathbb{R}^n} & \langle U \rangle \\ \text{such that} & \mathbf{x}^{(i)} - \mathbf{x}^{(i+1)} = 0 \end{array} \tag{1}$$

with particle states $\mathbf{x}^{(i)} \in \mathbb{R}^n$ for $i = 1, 2, ..., q$ and boundary conditions $\mathbf{x}^{(0)} = \mathbf{x}^{(q)}$, $\mathbf{x}^{(q+1)} = \mathbf{x}^{(1)}$. The synchronization constraints have to be achieved in an asymptotical sense, i.e. the particles have to reach the same final state.

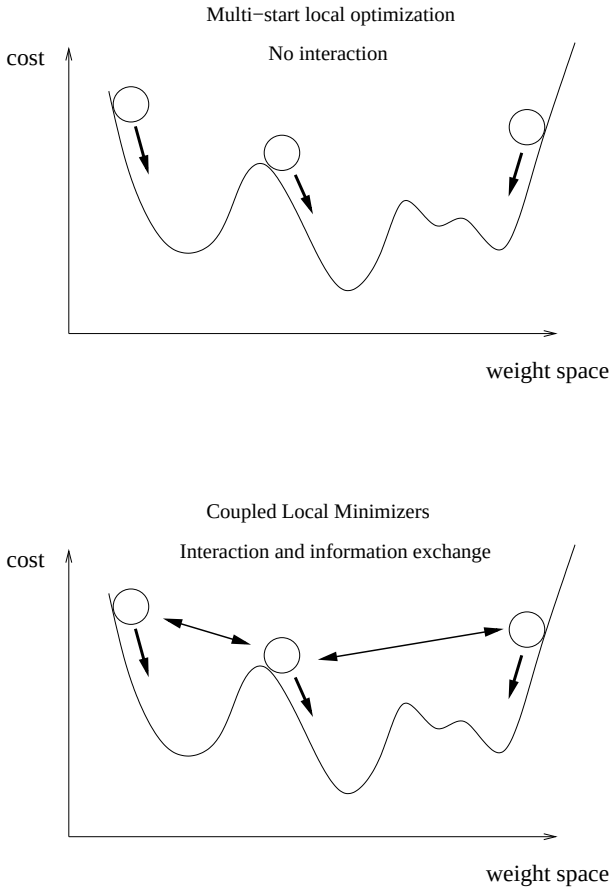


Figure 2: *In CLMs one considers interaction between local search processes instead of multi-start local optimization by independent runs.*

One defines the augmented Lagrangian [6]

$$\mathcal{L}(\mathbf{x}^{(i)}; \lambda^{(i)}) = \frac{\eta}{q} \sum_{i=1}^{q} U[\mathbf{x}^{(i)}] + \frac{1}{2} \sum_{i=1}^{q} \gamma_i \left\| \mathbf{x}^{(i)} - \mathbf{x}^{(i+1)} \right\|_2^2 + \sum_{i=1}^{q} \langle \lambda^{(i)}, [\mathbf{x}^{(i)} - \mathbf{x}^{(i+1)}] \rangle \tag{2}$$

with Lagrange multipliers $\lambda^{(i)} \in \mathbb{R}^n$ $(i = 1, 2, ..., q)$. The penalty factors γ_i emphasize the importance of each of the soft synchronization constraints.

From (2), one obtains the Lagrange programming network [12]

$$\left\{ \begin{array}{lcl} \dot{\mathbf{x}}^{(i)} & = & -\nabla_{\mathbf{x}^{(i)}} \mathcal{L}(\mathbf{x}^{(i)}; \lambda^{(i)}) \\ \dot{\lambda}^{(i)} & = & \nabla_{\lambda^{(i)}} \mathcal{L}(\mathbf{x}^{(i)}; \lambda^{(i)}) \,, \; i = 1, 2, ..., q. \end{array} \right. \tag{3}$$

This gives the following array of coupled local minimizers

(CLM)

$$\left\{\begin{array}{rcl} \dot{\mathbf{x}}^{(i)} & = & -\frac{\eta}{q}\nabla_{\mathbf{x}^{(i)}}U[\mathbf{x}^{(i)}]+\gamma_{i-1}[\mathbf{x}^{(i-1)}-\mathbf{x}^{(i)}]- \\ & & \gamma_i[\mathbf{x}^{(i)}-\mathbf{x}^{(i+1)}]+\lambda^{(i-1)}-\lambda^{(i)} \\ \dot{\lambda}^{(i)} & = & \mathbf{x}^{(i)}-\mathbf{x}^{(i+1)}\,,\; i=1,2,...,q \end{array}\right. \tag{4}$$

with learning rate η. This array consists of a number of q coupled nonlinear cells and can be considered as a cellular neural network (CNN) [3, 4].

3 Cooperative search and master-slave synchronization principles

An optimal interaction between the local minimizers can be obtained by making a suitable choice of the penalty weights γ_i. Suppose one is interested in maximally decreasing the instantaneous average energy cost of the ensemble [9]

$$\begin{aligned} \frac{d\langle U\rangle}{dt} &= \frac{1}{q}\sum_{i=1}^{q}\langle\frac{\partial U[\mathbf{x}^{(i)}]}{\partial\mathbf{x}^{(i)}},\dot{\mathbf{x}}^{(i)}\rangle \\ &= \frac{1}{q}\sum_{i=1}^{q}\langle\frac{\partial U[\mathbf{x}^{(i)}]}{\partial\mathbf{x}^{(i)}},-\frac{\eta}{q}\frac{\partial U[\mathbf{x}^{(i)}]}{\partial\mathbf{x}^{(i)}}+\gamma_{i-1}[\mathbf{x}^{(i-1)}-\mathbf{x}^{(i)}] \\ &\qquad -\gamma_i[\mathbf{x}^{(i)}-\mathbf{x}^{(i+1)}]+\lambda^{(i-1)}-\lambda^{(i)}\rangle. \end{aligned} \tag{5}$$

For a given state vector $\mathbf{x} = [\mathbf{x}^{(1)};\mathbf{x}^{(2)};...;\mathbf{x}^{(q)}]$ and costate of the ensemble $\lambda = [\lambda^{(1)};\lambda^{(2)};...;\lambda^{(q)}]$ the expression (5) is affine in $\gamma = [\gamma_1;\gamma_2;...;\gamma_q]$. The interaction between the local minimizers is optimized then by solving the linear program (LP) [6]:

$$\min_{\gamma\in\mathbb{R}^q}\frac{d\langle U\rangle}{dt}\,|_{\mathbf{x},\lambda}\quad\text{such that}\quad\underline{\gamma}<\gamma_i<\overline{\gamma} \tag{6}$$

where $\underline{\gamma},\overline{\gamma}$ are user-defined lower- and upperbounds. The resulting γ_i values are kept constant for simulation of the CLM over a certain time interval ΔT.

By applying this method in addition to the CLM coupling mechanism, speed-ups can be obtained by this parallel computing array, which is relevant in view of issues of complexity and real computation [1] and coherence in quantum computing [7].

The values of γ_i,η are re-scheduled each time in between simulations of the array over user-defined time intervals ΔT. According to the idea of small-world networks [10], the information spreading within the ensemble can be further improved, e.g. by re-numbering a part of the ensemble (state vectors together with their corresponding co-state) at random every $c\Delta T$ steps where $c\in\mathbb{N}_0$. In this way the pairwise synchronization constraints vary among the different local minimizers during optimization.

The difference between the γ_i values in (4) causes a similar effect as is known in master-slave (drive-response) synchronization which has been studied for secure communications using chaos [2, 8, 11]. Consider e.g. two dynamical systems with master $\mathcal{M}$ and slave system $\mathcal{S}$:

$$\left\{\begin{array}{rcl} \mathcal{M}: & \dot{\mathbf{x}} & = \mathbf{f}(\mathbf{x}) \\ \mathcal{S}: & \dot{\mathbf{z}} & = \mathbf{f}(\mathbf{z})+\mathbf{K}(\mathbf{x}-\mathbf{z}) \end{array}\right. \tag{7}$$

with state vectors $\mathbf{x},\mathbf{z}\in\mathbb{R}^m$ and feedback matrix $\mathbf{K}\in\mathbb{R}^{m\times m}$. In certain applications the systems are chaotic and the states converge to each other starting from different initial conditions. In this case the master system $\mathcal{M}$ imposes its state to the slave system $\mathcal{S}$. In order to see the link with the CLM array method (4), it suffices to write (7) as a scheme with mutual synchronization

$$\left\{\begin{array}{rcl} \mathcal{M}: & \dot{\mathbf{x}} & = \mathbf{f}(\mathbf{x})+\mathbf{0}\,(\mathbf{x}-\mathbf{z}) \\ \mathcal{S}: & \dot{\mathbf{z}} & = \mathbf{f}(\mathbf{z})+\mathbf{K}\,(\mathbf{x}-\mathbf{z}). \end{array}\right. \tag{8}$$

From the coefficients of the coupling terms one can see which system is imposing its state to the other.

4 Alternative CLM formulation

The original CLM formulation is done in terms of equality constraints. This leads to a co-state vector of the same size as the CLM state vector. We present now an alternative formulation in terms of inequality constraints which will results in a reduced size of the co-state vectors.

Consider the following problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}^{(i)}\in\mathbb{R}^n,r\in\mathbb{R}} & \langle U\rangle\;+\frac{1}{2}\,\nu\,r^2 \\ \text{such that} & \|\mathbf{x}^{(i)}-\mathbf{x}^{(i+1)}\|_2^2\le r^2,\quad i=1,...,q. \end{array} \tag{9}$$

Here ν denotes a positive real constant. After taking slack variables δ_i the inequality constraints are reformulated as $\|\mathbf{x}^{(i)}-\mathbf{x}^{(i+1)}\|_2^2+\delta_i^2=r^2$ which leads to the following Lagrangian

$$\begin{aligned} \mathcal{L}(\mathbf{x}^{(i)},r,\delta_i;\mu_i) &= \frac{\eta}{q}\sum_{i=1}^{q}U[\mathbf{x}^{(i)}]+\frac{1}{2}\,\nu\,r^2+\sum_{i=1}^{q}\mu_i\,\{ \\ &\|\mathbf{x}^{(i)}-\mathbf{x}^{(i+1)}\|_2^2+\delta_i^2-r^2\}+\frac{1}{2}\sum_{i=1}^{q}\gamma_i\,\|\mathbf{x}^{(i)}-\mathbf{x}^{(i+1)}\|_2^2 \end{aligned} \tag{10}$$

which is augmented with the same soft constraint terms as in (2), where μ_i denote the Lagrange multipliers. The

continuous time optimization algorithm becomes

$$\begin{cases} \dot{\mathbf{x}}^{(i)} &= -\nabla_{\mathbf{x}^{(i)}} \mathcal{L}(\mathbf{x}^{(i)}, r, \delta_i; \mu_i) \\ \dot{r} &= -\nabla_r \mathcal{L}(\mathbf{x}^{(i)}, r, \delta_i; \mu_i) \\ \dot{\delta}_i &= -\nabla_{\delta_i} \mathcal{L}(\mathbf{x}^{(i)}, r, \delta_i; \mu_i) \\ \dot{\mu}_i &= \nabla_{\mu_i} \mathcal{L}(\mathbf{x}^{(i)}, r, \delta_i; \mu_i) \end{cases} \tag{11}$$

which gives

$$\begin{cases} \dot{\mathbf{x}}^{(i)} &= -\frac{\eta}{q} \nabla_{\mathbf{x}^{(i)}} U[\mathbf{x}^{(i)}] + (\gamma_{i-1} + \mu_{i-1})[x^{(i-1)} - x^{(i)}] \\ & \quad -(\gamma_i + \mu_i)[x^{(i)} - x^{(i+1)}] \\ \dot{r} &= -(\nu + 2\sum_{i=1}^{q} \mu_i) r \\ \dot{\delta}_i &= -2\mu_i \delta_i \\ \dot{\mu}_i &= \|x^{(i)} - x^{(i+1)}\|_2^2 + \delta_i^2 - r^2. \end{cases} \tag{12}$$

Note that the number of equations in (12) can be significantly reduced with respect to (4), namely $qn + 2q + 1$ versus $2qn$, respectively, when $q \ll n$. The scheduling of the γ_i values can be done in a similar way as before by

$$\min_{\gamma \in \mathbb{R}^q} \frac{d\langle U \rangle}{dt} |_{\mathbf{x},\mu} \quad \text{such that} \quad \underline{\gamma} < \gamma_i < \overline{\gamma}. \tag{13}$$

In principle, many other alternative approaches might be followed to incorporate the constraints [5, 6] in (9).

5 Full versus sparse interaction scheme

In the present CLM schemes (1,9), pairwise state synchronization constraints have been considered. However, more complex interaction schemes could be taken. One may formulate

$$\begin{aligned} \min_{\mathbf{x}^{(i)} \in \mathbb{R}^n, r \in \mathbb{R}} \quad & \langle U \rangle \ + \frac{1}{2} \nu r^2 \\ \text{such that} \quad & \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2^2 \leq r^2, \end{aligned} \tag{14}$$

e.g. for $i, j = 1, ..., q$ or for $i > j \, (j = 1, .., q)$. Such a scheme might offer additional degrees of freedom concerning the soft penalty terms in the augmented Lagrangian (more γ_i values due to additional constraints) which could be employed in order to improve (13). On the other hand, the number of equations to be simulated will increase.

6 Illustrative example

Let us illustrate the alternative CLM formulation on the following example with cost function

$$U(\mathbf{x}) = \frac{1}{2n} a \sum_{i=1}^{n} x_i^2 - 4n \prod_{i=1}^{n} \cos(\omega x_i) \tag{15}$$

with $a = 1$, $\omega = 1$ and a $n = 10$ dimensional search space. Fig.3-4 show simulation examples on a comparison between multistart local optimization (i.e. steepest descent without coupling) versus the CLM approach (12). The same initial conditions were taken (normal distribution with zero mean and standard deviation equal to 3). The constant $\nu = 1$ was taken and $\eta = 1, \gamma_i = 10 \, \forall i$.

The benefit of realizing the coupling is clear from the simulations. The global minimum is achieved by the CLM while not by the multistart local optimization. This has been verified by taking many different random initial conditions. In these simulations the basic alternative CLM scheme has been applied. The performance can be further enhanced by incorporating the ideas of Section 3 and taking an adaptive learning rate.

7 Conclusion

In CLMs one aims at coupling local optimization methods in order to create interaction and information exchange. We have discussed how one can let such optimizers interact with each other in order to obtain improved results. An alternative formulation is given in this paper in order to obtain a CLM in less equations to be simulated. By means of the CLM method one can allow exploration of the search space and realize speed-ups which is important in view of fundamental problems of complexity of real machines [1] and in relation to speed-ups obtained by quantum computers [7]. The CLM method offers a new theoretical framework towards space-time complexity of CNNs in general.

Acknowledgments. This research work was carried out at the ESAT laboratory and the Interdisciplinary Center of Neural Networks ICNN of the Katholieke Universiteit Leuven, in the framework of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture (IUAP P4-02 & IUAP P4-24), the Concerted Action Project MEFISTO of the Flemish Community and the FWO project *Collective Behavior and Optimization: an Interdisciplinary Approach.* Johan Suykens is a postdoctoral researcher with the National Fund for Scientific Research FWO - Flanders.

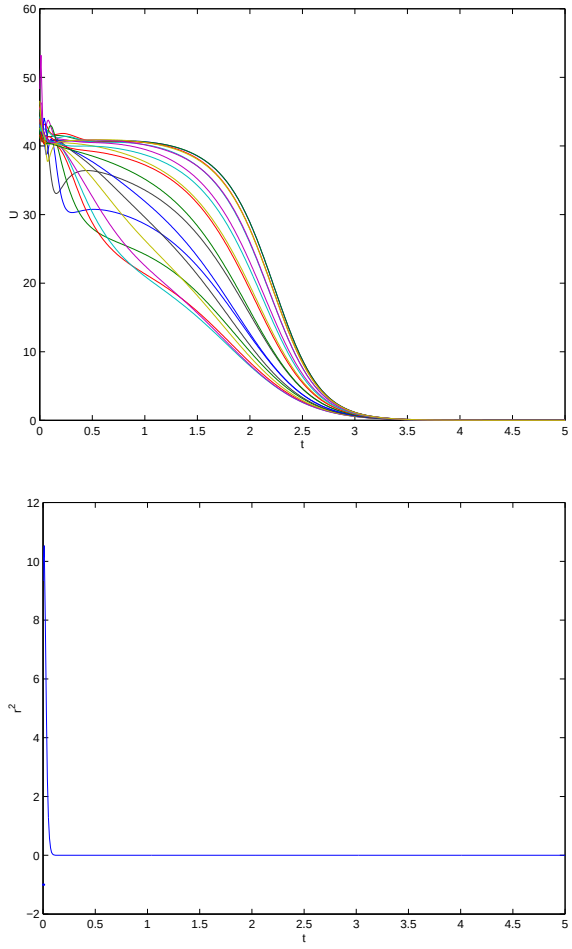


Figure 3: *Illustration of CLM convergence with $q = 20$ particles: (Top) $U[\mathbf{x}^{(i)}]$ with respect to time t; (Bottom) r^2 with respect to time t.*

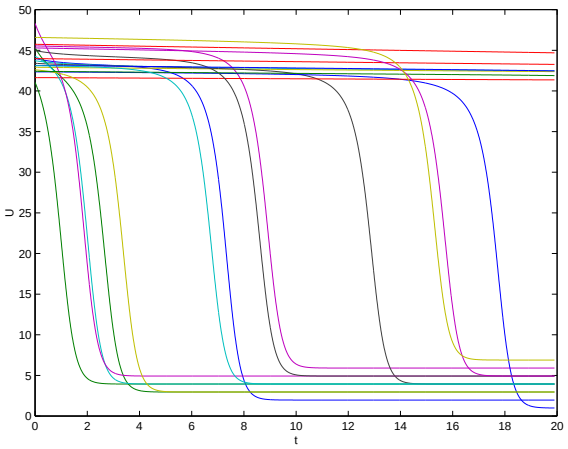


Figure 4: *Multistart local optimization for the same problem as in Fig.1 with the same initial conditions but without coupling. Shown are $U[\mathbf{x}^{(i)}]$ with respect to time t.*

References

[1] Blum L., Cucker F., Shub M., Smale S., *Complexity and Real Computation,* Springer-Verlag, 1997.

[2] Chen G., Dong X., *From Chaos to Order - Perspectives, Methodologies, and Applications*, World Scientific Pub. Co., Singapore, 1998.

[3] Chua L.O., Roska T., "The CNN Paradigm," *IEEE Trans. Circuits and Systems-I*, **40**(3), 147-156, 1993.

[4] Chua L.O., *CNN: a Paradigm for Complexity*, World Scientific Pub. Co., Signapore, 1998.

[5] Cichocki A., Unbehauen R., *Neural Networks for Optimization and Signal Processing*, Wiley, Chichester, 1994.

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

[7] Shor P.W., "Quantum Computing," *Doc. Math. J. DMV Extra Volume ICM I*, 467-486, 1998.

[8] Suykens J.A.K., Vandewalle J., "Chaos synchronization: a Lagrange programming network approach," *Int. J. Bifurcation and Chaos (special issue on chaos control and synchronization)*, **10**(4), 797-810, 2000.

[9] Suykens J.A.K., Vandewalle J., De Moor B., "Intelligence and Cooperative Search by Coupled Local Minimizers", *Int. J. Bifurcation and Chaos*, **11**(8), 2133-2144, 2001.

[10] Watts D.J., Strogatz S.H. , "Collective dynamics of 'small-world' networks," *Nature* **393**, 440-442, 1998.

[11] Wu C.W., Chua L.O., "A unified framework for synchronization and control of dynamical systems," *Int. J. Bifurcation and Chaos* **4**, 979-989, 1994.

[12] Zhang S., Constantinides A.G., "Lagrange programming neural networks," *IEEE Trans. Circuits and Systems-II* **39**, 441-452, 1992.