5-2010

Bio-Inspired Distributed Constrained Optimization Technique and its Application in Dynamic Thermal Management

Saranya Chandrasekaran
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd
Part of the Computer Engineering Commons

Recommended Citation
https://digitalcommons.usu.edu/etd/701
BIO-INSPIRED DISTRIBUTED CONSTRAINED OPTIMIZATION TECHNIQUE 
AND ITS APPLICATION IN DYNAMIC THERMAL MANAGEMENT

by

Saranya Chandrasekaran

A thesis submitted in partial fulfillment 
of the requirements for the degree 
of 
MASTER OF SCIENCE 
in 
Computer Engineering

Approved:

Dr. Aravind Dasu 
Major Professor 

Dr. David Peak 
Committee Member 

Prof. Paul Israelsen 
Committee Member 

Dr. Byron R. Burnham 
Dean of Graduate Studies 

UTAH STATE UNIVERSITY  
Logan, Utah  
2010
Abstract

Bio-Inspired Distributed Constrained Optimization Technique and its Application in Dynamic Thermal Management

by

Saranya Chandrasekaran, Master of Science
Utah State University, 2010

Major Professor: Dr. Aravind Dasu
Department: Electrical and Computer Engineering

The stomatal network in plants is a well-characterized biological system that hypothetically solves the constrained optimization problem of maximizing CO$_2$ uptake from the air while constraining evaporative water loss during the process of photosynthesis. There are numerous such constrained optimization problems present in the real world as well as in computer science. This thesis work attempts to solve one such constrained optimization problem in a distributed manner by taking a cue from the dynamics of stomatal networks. The problem considered here is Dynamic Thermal Management (DTM) in a multi-processing element system in computing. There have been several approaches in the past that tried to solve the problem of DTM by varying the frequency of operation of blocks in the computing system. The selection of frequencies for DTM such that overall performance is maximized while temperature is constrained is a non-deterministic polynomial-time (NP) hard problem. In this thesis, a distributed approach to solve the problem of DTM using a cellular neural network is proposed. A cellular neural network is used to mimic the stomatal network with slight variations based on the problem considered.
To my parents.
Acknowledgments

I would like to thank Dr. Aravind Dasu, my advisor, for being a constant motivator and for his help during the course of my master’s program. I would like to thank Dr. David Peak for his guidance and help in understanding several concepts in this thesis work.

I would like to thank my lab-mates, Varun Voddi and Ramachandra Kallam, for their help in my research. I thank all my friends who have been of constant moral support throughout my studies. I would like to thank my parents and brother for their love and support.

Saranya Chandrasekaran
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Constrained Optimization Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Key Contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Thesis Overview</td>
<td>3</td>
</tr>
<tr>
<td>2 Background</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Common Optimization Problems</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1 Graph Theory Problems</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2 Packing Problems</td>
<td>4</td>
</tr>
<tr>
<td>2.1.3 Scheduling Problems</td>
<td>5</td>
</tr>
<tr>
<td>2.1.4 Nonlinear Least Square Problems</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Methods to Solve Optimization Problems</td>
<td>6</td>
</tr>
<tr>
<td>2.2.1 Adaptive Stochastic Search Methods</td>
<td>6</td>
</tr>
<tr>
<td>2.2.2 Branch and Bound Algorithms</td>
<td>6</td>
</tr>
<tr>
<td>2.2.3 Enumerative Strategies</td>
<td>6</td>
</tr>
<tr>
<td>2.2.4 Homotopy and Trajectory Methods</td>
<td>7</td>
</tr>
<tr>
<td>2.2.5 Relaxation Strategies</td>
<td>7</td>
</tr>
<tr>
<td>2.2.6 Approximate Convex Underestimation</td>
<td>7</td>
</tr>
<tr>
<td>2.2.7 Genetic Algorithms, Evolution Strategies</td>
<td>7</td>
</tr>
<tr>
<td>2.2.8 Simulated Annealing</td>
<td>8</td>
</tr>
<tr>
<td>2.2.9 Globalized Extensions of Local Search Methods</td>
<td>8</td>
</tr>
<tr>
<td>2.2.10 Stochastic Tunneling</td>
<td>9</td>
</tr>
<tr>
<td>2.2.11 Swarm-Based Optimization Techniques</td>
<td>9</td>
</tr>
<tr>
<td>2.2.12 Tabu Search</td>
<td>9</td>
</tr>
<tr>
<td>3 Stomatal Network and Cellular Neural Network</td>
<td>11</td>
</tr>
<tr>
<td>3.1 Stomatal Network</td>
<td>11</td>
</tr>
<tr>
<td>3.1.1 Stomata</td>
<td>11</td>
</tr>
<tr>
<td>3.1.2 Stomatal Patchiness</td>
<td>13</td>
</tr>
<tr>
<td>3.1.3 Stomatal Dynamics</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Cellular Neural Network</td>
<td>15</td>
</tr>
</tbody>
</table>
4 DTM ................................................................. 21
   4.1 Motivation ..................................................... 21
   4.2 DTM: Overview and Strategies ................................ 22
      4.2.1 Temperature Trigger Techniques ....................... 22
      4.2.2 DTM: Response Techniques .............................. 24
   4.3 DTM: Problem Formulation .................................. 27

5 Bio-Inspired DTM Technique ........................................ 29
   5.1 DTM Model .................................................... 29
      5.1.1 System of Equations ..................................... 29
      5.1.2 Temperature Modeling ................................. 30
      5.1.3 Steady-State Analysis ................................. 31
   5.2 Mapping Cellular Neural Network and Real-Time System .......... 33
   5.3 Bio-Inspired Algorithm for DTM .............................. 35

6 Results ............................................................. 39
   6.1 Weight Determination ........................................ 39
   6.2 Adaptive Response .......................................... 41
   6.3 Thermal Profile ............................................ 41
   6.4 Advantages .................................................. 45

7 Conclusions and Future Work ....................................... 46

References .......................................................... 48
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Cellular computing networks.</td>
<td>20</td>
</tr>
<tr>
<td>5.1</td>
<td>Stomatal network and cellular neural network for DTM.</td>
<td>38</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Genetic algorithm.</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Simulated annealing algorithm.</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Tabu search algorithm.</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>Stomatal network.</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Stoma.</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>Stomatal patchiness.</td>
<td>14</td>
</tr>
<tr>
<td>3.4</td>
<td>Stomatal plug network.</td>
<td>16</td>
</tr>
<tr>
<td>3.5</td>
<td>A cellular neural network.</td>
<td>16</td>
</tr>
<tr>
<td>3.6</td>
<td>A cellular neural network cell/node circuit.</td>
<td>16</td>
</tr>
<tr>
<td>3.7</td>
<td>Cellular neural network nonlinear output function.</td>
<td>18</td>
</tr>
<tr>
<td>4.1</td>
<td>Response of DTM technique.</td>
<td>23</td>
</tr>
<tr>
<td>4.2</td>
<td>HotSpot thermal model - complete and lateral.</td>
<td>24</td>
</tr>
<tr>
<td>4.3</td>
<td>Multiple clock domains in Xilinx FPGA.</td>
<td>25</td>
</tr>
<tr>
<td>5.1</td>
<td>DTM - cellular neural network.</td>
<td>30</td>
</tr>
<tr>
<td>5.2</td>
<td>Simplified RC model.</td>
<td>32</td>
</tr>
<tr>
<td>5.3</td>
<td>Thermal model simulation.</td>
<td>32</td>
</tr>
<tr>
<td>5.4</td>
<td>High-level block diagram of the system.</td>
<td>34</td>
</tr>
<tr>
<td>5.5</td>
<td>Example configurations.</td>
<td>35</td>
</tr>
<tr>
<td>5.6</td>
<td>Stomatal network and cellular neural network.</td>
<td>36</td>
</tr>
<tr>
<td>5.7</td>
<td>Bio-inspired DTM algorithm flowchart.</td>
<td>37</td>
</tr>
<tr>
<td>6.1</td>
<td>Weights determination.</td>
<td>40</td>
</tr>
<tr>
<td>6.2</td>
<td>Adaptive response.</td>
<td>42</td>
</tr>
<tr>
<td>6.3</td>
<td>Numerical results.</td>
<td>43</td>
</tr>
<tr>
<td>6.4</td>
<td>Temperature simulation.</td>
<td>44</td>
</tr>
<tr>
<td>6.5</td>
<td>Steady-state convergence of temperature.</td>
<td>44</td>
</tr>
</tbody>
</table>
Chapter 1
Introduction

In this chapter, an introduction to constrained optimization problems is presented. The motivation for developing a Bio-Inspired Distributed Constrained Optimization technique is discussed, followed by the key contributions of this thesis. This is followed by an overview of the content presented in this thesis.

1.1 Constrained Optimization Problem

In constrained optimization problem, the objective is to find a solution whose cost, evaluated as a sum of cost functions, is maximized or minimized, while the constraints are satisfied. The regular constraints are called hard constraints, while the cost functions are called soft constraints.

Mathematically, a constrained optimization problem is one for which a function $f(x)$ is to be minimized or maximized subject to constraints $\phi(x)$. Here $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is called the objective function and $\phi(x)$ is a boolean-valued formula. The constraints $\phi(x)$ can be an arbitrary boolean combinations of equations $g(x)=0$, weak inequalities $g(x)\geq 0$, strict inequalities $g(x)>0$, and $x\in \mathbb{Z}$ statements. A constrained optimization problem is usually represented as Min. $f(x)$ s.t. $\phi(x)$ or Max. $f(x)$ s.t. $\phi(x)$.

1.2 Motivation

Many real-world problems are mathematically modeled to be constrained optimization problems and can be utilized for a very broad range of applications. Applications of constrained optimization problems include structural optimization, engineering design, very-large-scale integration (VLSI) chip design, database problems, nuclear and mechanical design, chemical engineering, economies of scale, allocation and location problems, packing
problems (knapsack problem), scheduling problems, protein folding, chemical equilibrium, robotic control, and a number of other combinatorial optimization problems such as integer programming and graph problems (maximum clique problem) [1].

Such optimization problems generally fall under the class of non-deterministic polynomial-time (NP) hard problems. This means that as the input size of the problem increases, the computational time required grows exponentially. Active research in this area has produced a variety of heuristic methods for determining global solutions for continuous constrained optimization problems (discussed in Chapter 2).

The key factors to approaches solving such problems are the optimality of results obtained and speed of the algorithm. A parallel approach inspired by plant system dynamics (stomatal network) for solving such constrained optimization problems is proposed in this thesis. The thesis explores the plant computing model and the continuous optimization problem of maximizing \( CO_2 \) uptake for a constrained amount of \( H_2O \) loss solved by stomatal network. A similar problem in computation, Dynamic Thermal Management (DTM), is studied, which is a constrained optimization problem where the goal is to maximize performance while constraining the temperature to a working level. With the advent of multi-processing elements on a single chip for achieving high performance, the challenges with respect to reducing the power dissipation and maintaining the chip temperature at working level by not affecting the overall performance have become all the more significant, thus, motivating researchers to find solutions that would be extremely fast, accurate, and manage chip temperatures on the fly. This thesis explores the similarity between the two optimization problems discussed above and use the knowledge of the stomatal network dynamics to solve the DTM problem.

1.3 Key Contributions

The key contributions of this thesis are:

1. A model to solve continuous constrained optimization problem based on stomatal dynamics,
2. A cellular neural network weight calculation technique using steady-state analysis,
3. A mathematical formulation of the DTM problem, and

1.4 Thesis Overview

The plant computing model, stomatal network dynamics, and background work related towards categorizing and solving constrained optimization problems are presented in Chapter 2. Chapter 3 gives the plant computational model and an introduction to cellular neural networks and other cellular networks which are circuits that can be used to mimic the behavior found in stomatal network. Chapter 4 gives an introduction to the DTM problem and related work. In Chapter 5, the distributed constrained optimization technique based on stomatal network is discussed for the DTM problem. A comparative study on the plant system and the proposed technique is presented. Results of the proposed method is presented in Chapter 6. Chapter 7 gives the conclusions and the future work.
Chapter 2

Background

In this chapter, classification of constrained optimization problems is discussed, followed by a review on methods developed in the past for solving such problems.

2.1 Common Optimization Problems

As discussed in sec. 1.2, there are several examples of constrained optimization problems in applied mathematics, computing and real life. Broadly, these optimizations can be categorized based on their applications [2].

2.1.1 Graph Theory Problems

Many graph theory problems are optimization problems, the maximum clique problem is a typical example for an optimization problem in which the maximum number of mutually adjacent vertices in a given graph needs to be found. This problem has applications in bioinformatics [3, 4], computational chemistry, VLSI design automation [5], and real-world social networks [6]. A mathematical formulation for this optimization is

\[ \max x^T Ax \]
\[ s.t. \quad e^T x = 1, \quad x \geq 0, \]

(2.1)

where A is the adjacency matrix of the graph and e is an all-one vector. This formulation is an indefinite quadratic program [7].

2.1.2 Packing Problems

The simplest of packing problems is the knapsack problem. In the knapsack problem, a maximal number of objects of given weights is to be placed into a knapsack with a contraint
on maximum weight capacity of knapsack. If there are \( n \) kinds of items, and each item \( i \) has a value \( v_i \), weight \( w_i \) and the maximum capacity is restricted by \( W \), then the mathematical formulation of the problem is

\[
\begin{align*}
\max & \quad \sum v_i x_i \\
\text{s.t.} & \quad \sum w_i x_i \leq W, \ x_i \in \{0, 1\} \text{ or } x_i \in \{0, 1, ..., c_i\} \text{ or } x_i \in \{0, 1, ..., \infty\}.
\end{align*}
\] (2.2)

There are several applications of the knapsack problem to auction design and related bidding problem [8], VLSI [9], industrial and several practical applications [10].

### 2.1.3 Scheduling Problems

Largely, scheduling problems target matching tasks and slots of execution such that every task is handled, total cost is minimized/maximized and some additional constraints are satisfied. The simplest formulation of scheduling problem is the sequencing of tasks (with release time and deadlines) on a single processor. For a set \( T \) of tasks, where each task \( t \in T \) takes \( l(t) \in \mathbb{Z}^+ \), and has a release time of \( r(t) \in \mathbb{Z}_0^+ \) and deadline \( d(t) \in \mathbb{Z}^+ \); the problem is to find a one-processor schedule for \( T \) such that all release time constraints and task deadlines are met, i.e., to find a function

\[
\sigma : T \rightarrow \mathbb{Z}_0^+
\]

\[
\text{s.t.} \quad \sigma(t) \geq r(t) \text{ and } \sigma(t) + l(t) \leq d(t).
\] (2.3)

Variants of the scheduling problem include multi-processor scheduling, resource constrained scheduling, job-shop scheduling, staff scheduling, production planning [11].

### 2.1.4 Nonlinear Least Square Problems

Fitting a function to a set of data points is often required in many applications related to artificial intelligence. This leads to an optimization problem where the objective function is of the form as shown in eq. (2.4) where \( x, y \) are given data vectors and \( \theta \) is a parameter
vector. The problem is to find values of $\theta$ such that $f$ is minimized.

\[ f(\theta) = \sum ||y_i - F(x_i, \theta)||^2 \]  \hfill (2.4)

This problem mainly applies to machine learning, self correcting artificial neural networks, and curve fitting, and has practical applications in forecasting, finance, etc.

2.2 Methods to Solve Optimization Problems

Many methods have been developed in the past to solve constrained optimization problems. These cover both exact and heuristic methods commonly used to solve both discrete and continuous global optimization problems.

2.2.1 Adaptive Stochastic Search Methods

The adaptive stochastic search methods depends on partial random sampling in the solution space. These methods are improved by adding strategic adjustments, solution refinement methods, rejection rules to the pure random sampling concept. These methods are used for both discrete and continuous optimization problems [12].

2.2.2 Branch and Bound Algorithms

With branch and bound algorithms, the solution set is split first into subsets, which is called the branching step, followed by another procedure called bounding estimates the upper and lower bounds of solutions based on the objective function. The calculation of bounds in each subset helps in discarding solution sets while searching for the optimal solution, which is called pruning. The algorithm is recursive and stops when the optimal value is reached. Adaptive partitioning, sampling and bounding procedures are applied in this technique for continuous global optimizations [13].

2.2.3 Enumerative Strategies

Enumerative strategies are based on complete enumeration of all possible solutions of
a given problem. They are constituted by variable and value selection heuristics. Selection strategies can be static or dynamic [14]. These methods are more useful in combinatorial optimization problems [15].

2.2.4 Homotopy and Trajectory Methods

Homotopy and trajectory methods target at visiting all stationary points of a given objective within the solution space. The result of this search is the list of all global and local optimal points in the search space. These methods are useful for smooth global optimization problems.

2.2.5 Relaxation Strategies

Relaxation approach depends on breaking down the optimization problem into a sequence of relaxed sub-problems that are easier to solve. Refinement of the sub-problems to approximate the original problem is done. In this general approach, the optimization problem is replaced by a sequence of relaxed sub-problems that are easier to solve. Relaxation algorithms are applicable to optimization problems such as concave minimization, discrete choice dynamic programming models, mixed integer nonlinear problems [16].

2.2.6 Approximate Convex Underestimation

The convex character of the objective function is estimated based on direct sampling from the solution space. Convex underestimation approximation is used to solve signomial functions [17], continuously differentiable functions [18], and smooth optimization problems.

2.2.7 Genetic Algorithms, Evolution Strategies

Genetic algorithm is a heuristic search technique used for finding exact or approximate solutions by mimicking biological evolutionary models. Various stochastic algorithms can be derived based on diverse evolutionary rules which can be applied to both discrete and continuous optimization problems. In genetic algorithm, a population of chromosomes (solutions) is considered. New populations of chromosomes are produced based on current
population through reproduction (mutation/recombination). The survival of the fittest rule is applied to retain the best solutions and the cycle is carried on until the best solution is obtained. Figure 2.1 shows the top-level description of a simple genetic algorithm [19].

2.2.8 Simulated Annealing

The simulated annealing technique is based upon the physical analogy of cooling crystal structures that spontaneously arrive at a stable configuration that is characterized by minimal potential energy (global or local) [20].

Simulated annealing is applicable to both discrete and continuous optimization problems. This algorithm is based on perturbing current solutions and probabilistically accepting new solutions so as to avoid reaching local minima or maxima and search for global optimum solution. An initial high temperature is set and algorithm is carried on (cooling period) till a threshold temperature is reached (refer to fig. 2.2) [21].

2.2.9 Globalized Extensions of Local Search Methods

Globalized extensions of local search methods methods are based on initial global search followed by local scope search. Such techniques are applicable to smooth optimization problems. They are partial heuristic techniques which employ grid or random search-based global search.

Fig. 2.1: Genetic algorithm.
2.2.10 Stochastic Tunneling

Stochastic tunneling is a Monte Carlo sampling-based approach for global optimization. Monte Carlo technique involves sampling the objective function by randomly hopping from a solution vector to another with a slight difference in the function value with a determined acceptance probability. The algorithm mimics the tunneling of barriers in a system having a complex energy function [22].

2.2.11 Swarm-Based Optimization Techniques

Swarm-based algorithms are multi-agent approaches for solving optimization problems. The system constitutes of simple agents that interact locally with each other and their environment and follow simple rules without a centralized controller [23]. Examples of such algorithms are ant-colony, particle swarm, intelligent water drops, etc.

2.2.12 Tabu Search

Tabu search is a metaheuristic algorithm which is based on the idea of forbidding search moves to points already visited in the search space. Tabu search algorithm is mainly used for solving combinatorial optimization problems. The algorithm is shown in fig. 2.3 [24].
Fig. 2.3: Tabu search algorithm.
Chapter 3

Stomatal Network and Cellular Neural Network

3.1 Stomatal Network

Many biological systems adaptively respond to external environmental stimuli in the absence of a central nervous system. These responsive behaviors of biological systems suggest the possibility of the presence of computational rules followed by them. This idea has been explored in the stomatal system [25]. Stomata are tiny variable aperture pores on the surface of a leaf. They appear to act like biological processing units whose function is to help the plant solve an optimization problem.

The problem solved by a plant through its stomata is a continuous constraint optimization problem. It maximizes \( CO_2 \) intake for a given amount of \( H_2O \) loss. It has been observed that under specific environmental conditions, groups of stomata coordinate in both space and time producing patches that can be visualized with chlorophyll fluorescence. Analysis of the dynamic behavior of these patches has shown that stomata behave collectively. They form a network that is hypothesized to be engaged in the optimization task.

Figure 3.1 is a microscope image of the surface of Vicia faba leaf showing stomata (bean shaped structures) and the neighboring epidermal cells. The stomatal pore apertures are about 2 \( \mu m \) wide [25].

3.1.1 Stomata

Stomata are pores on the surfaces of leaves that permit the exchange of gases between the inside of the leaf and the atmosphere. In many plants, stomata are between 30 and 60 \( \mu m \) long and occur at densities between 50 and 200/mm\(^2\). Figure 3.2 shows an image of two neighboring stomata. A stoma consists of two guard cells. The guard cells change
their shape due to changes in their internal water content level. This process, known as osmosis, creates a pore of variable aperture. Gas diffusion occurs through the open stomatal pores. CO$_2$ enters the leaf, which is essential for the process of photosynthesis to occur, and simultaneously water vapor escapes from the leaf. Excessive water loss can have disastrous consequences for a plant. Hence, plants are faced with a problem of determining the pore aperture for a given set of environmental conditions that optimizes CO$_2$ uptake subject to some constraint on water loss.

The turgor pressures in the guard and epidermal cells change in response to variations in light intensity and ambient concentrations of CO$_2$ and water vapor. The guard cells flex and bow with this change in turgor pressure and cause a gap between them to change in turn. It is well established that stomatal aperture tends to increase when (a) light is increased, (b) water vapor is increased, and (c) CO$_2$ is decreased.

Experiments done with isolated epidermes [26] in which stomata are uncoupled from one another show that the stomatal aperture that is wide open tends to close and vice versa for a given set of environmental inputs. Also, other experiments show that open stomata in the neighborhood of a closed stoma tend to cause that stoma to open and vice versa.
3.1.2 Stomatal Patchiness

In the traditional model of stomatal function, stomata are considered as autonomous units that respond independently to environmental inputs of light, $CO_2$ and $H_2O$. It is predicted that for considerably slow environmental changes, stomatal conductance $g$ which is determined by aperture, varies as environmental conditions change such that

$$\frac{\partial A}{\partial g} \propto \frac{\partial E}{\partial g}.$$  

(3.1)

Here, $A$ is the rate of $CO_2$ uptake and $E$ is the rate of water loss. The spatial distribution of $g$ is predicted to be uniform in a spatially uniform environment. There could be minor differences in $g$ based on small structural difference in stomata. However, on the contrary, experiments show that despite uniform environmental conditions, groups of tens to thousands of stomata can behave extremely differently from other stomata present in their immediate neighborhood [27]. This behavior in the stomatal network is called stomatal patchiness.

Figure 3.3 is a chlorophyll fluorescence image of a Xanthium Strumarium leaf. The open stomata appear as dark regions and closed stomata appear as light regions. The image contains over 100,000 stomata. This illustrates stomatal patchiness where a leaf is maintained at uniform spatial environmental conditions. The image is taken in the near infrared and in carefully controlled conditions. The chlorophyll fluorescence is interpreted.
as being inversely proportional to stomatal conductance. The stomatal patches, in rare cases, continue to exist for hours and display rich dynamics which suggests the existence of space-time systems with self organizing dynamics [28].

3.1.3 Stomatal Dynamics

A simple model is developed to illustrate how stomata respond to environmental humidity and CO$_2$ concentration. A stoma is a pair of guard cells $G$. Stomata are located on a square lattice and each stoma is surrounded by a plug of tissues, namely, epidermal cells $E$, mesophyll $M$, and a water source $S$ (fig. 3.4). When a light-activated signal triggers the opening of a stomatal pore and water vapor diffusion occurs. This diffusion of water vapor in turn draws water from the source. In this model, the pore aperture $a$ is determined by a weighted difference of turgor pressures in guard cells and neighboring epidermal cells:

$$a = f[\beta(P_g - mP_e)].$$  \hspace{2cm} (3.2)

In eq. (3.2), the function $f$ decides the value of aperture and limits the value between 0 and $a_m$. If $\beta(P_g - mP_e) \leq 0$, $a$ is set to 0; if $\beta(P_g - mP_e) \geq a_m$, $a$ is set to $a_m$; and returns $\beta(P_g - mP_e)$ otherwise. The value $m$ represents the mechanical advantage of the epidermal cells over the guard cells. In this model, $P_e$ and $P_g$ change in time according to
\[
\frac{dP_e}{dt} = k_e \left[ (-\delta_e \Delta wa - P_e + \pi_e) - c_e (P_e - \langle P_e \rangle) \right], \\
\frac{dP_g}{dt} = k_g \left[ -\delta_g \Delta wa - P_g + \pi_g \right].
\]

In eqs. (3.3) and (3.4), \(\delta\)s and ks are coupling and rate constants, the \(\pi\)s are the respective osmotic pressures at site (ij), \(\Delta w\) the difference between internal and external humidites, and \(\langle P_e \rangle\) is the average epidermal turgor pressure of the neighboring epidermal cells. The dependence of aperture on light and \(CO_2\) is contained in \(\pi_g\) and \(\pi_e\) and in a uniformly lit life is constant throughout the stomatal network. The interaction between neighboring stomatal units is represented by the \(c_e (P_e - \langle P_e \rangle)\) term in this model.

### 3.2 Cellular Neural Network

As defined above, a stomatal network is a kind of cellular neural network. Cellular neural network is a class of computational networks with cellular components.

Invented by Leon Chua and Lin Yang in 1988 [29], cellular neural networks are artificial neural networks in which the cells are only locally connected (see fig. 3.5). Each cell has both linear and nonlinear elements, which are typically linear resistors, capacitors, linear and nonlinear controlled current sources, and independent sources. An example cell circuit is shown in fig. 3.6. Though there could theoretically be a cellular neural network of any dimension, two-dimensional cellular neural networks are common. Cells are arranged as a grid of M rows and N columns constituting an MxN network.

The neighborhood for a cell, \(N(i,j)\), in a cellular neural network can be \(N = 1\) (3x3 neighborhood), \(N = 2\) (5x5 neighborhood), or \(N = 3\) (7x7 neighborhood) and so on. Input, the internal state and the output of cells \((u, x, \text{and} y, \text{respectively})\) follow the dynamics:
Fig. 3.4: Stomatal plug network.

Fig. 3.5: A cellular neural network.

Fig. 3.6: A cellular neural network cell/node circuit.
\[ C \frac{d x_{ij}(t)}{dt} = - \frac{1}{R_x} x_{ij}(t) + \sum A(i, j; k, l) y_{kl}(t) + \sum B(i, j; k, l) u_{kl}(t) + I, \]
\[(1 \leq i \leq M; 1 \leq j \leq N), \quad (3.5)\]
\[ y_{ij}(t) = \frac{1}{2} (|x_{ij}(t) + 1| - |x_{ij}(t) - 1|), \quad (3.6)\]
\[ |x_{ij}(0)| \leq 1, \quad (3.7)\]
\[ |u_{ij}| \leq 1, \quad (3.8)\]
\[ A(i, j; k, l) = A(k, l; i, j), \quad (3.9)\]
\[ C > 0, R_x > 0, \quad (3.10)\]

where the sums are over the cells in \( N(i, j) \) at sites \( (k, l) \). The output function in eq. (3.6) is the nonlinear function shown in fig. 3.7. It is assumed that all the variables are continuous. The matrices \( A \) and \( B \) are called the cloning templates and the values of these matrices define the operation performed by the network. In the simplest cellular neural networks, the cloning template does not depend on the position of the cell. It is proved [29] that the cellular neural network converges to a steady state when the cloning template meets the constraint specified in eq. (3.9).

There can be cellular neural networks with more than one state variable. These constitute the class of multi-layer cellular neural networks. In this case, \( u, x, \) and \( y \) are vectors of values. In such a network, information is passed from the lowest layer upwards. (Note: Multi-layer networks are useful to model stomatal networks where there is more than one state variable.) The dynamics of a multi-layer cellular neural network is given by the eq. (3.11).

\[ C \ast \frac{d x_{ij}(t)}{dt} = - \frac{1}{R_x} \ast x_{ij}(t) + A \ast y_{ij}(t) + B \ast u_{ij}(t) + I, \]
\[(1 \leq i \leq M; 1 \leq j \leq N) \quad (3.11)\]
where

$$C = \begin{bmatrix} C_1 & 0 & 0 \\ \vdots & \ddots & 0 \\ 0 & \vdots & C_m \end{bmatrix}$$

$$R = \begin{bmatrix} R_{1x} & 0 & 0 \\ \vdots & \ddots & 0 \\ 0 & \vdots & R_{mx} \end{bmatrix}$$

$$A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & 0 \\ \vdots & 0 & 0 & \vdots & \vdots \\ \vdots & \vdots & 0 & \vdots & \vdots \\ A_{m1} & \vdots & \vdots & A_{mm} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{11} & 0 & 0 & 0 & 0 \\ \vdots & 0 & 0 & \vdots & \vdots \\ \vdots & \vdots & 0 & \vdots & \vdots \\ B_{m1} & \vdots & \vdots & B_{mm} \end{bmatrix}$$

Fig. 3.7: Cellular neural network nonlinear output function.

$$f(v) = \frac{1}{2} \left[ |v + 1| - |v - 1| \right]$$
The stability of multi-layer cellular neural networks is proved from the bottom layer upwards by ensuring block triangular structures of connecting matrices A and B. The time constants for each layer of the multi-layer cellular neural network can be chosen differently based on the operation.

Cellular neural networks have an extremely parallel setup with ultra high frame rate. Cellular neural networks are used for modeling physical and biological systems, as well as to perform a specific task such as image processing [30]. The latter includes feature extraction, edge detection, pattern learning/ recognition, motion estimation, image encoding/decoding, contouring, etc. All applications are implemented by carefully choosing the values of the cloning template.

There are other cellular computing systems other than cellular neural networks. As a general rule, all cellular computing systems comprise of individual cells which are arranged in one-to-many-dimensional lattice. A cell is connected to only a subset of all cells in the network and state value for each cell is updated simultaneously based on the rule defined for the network. Cellular computing systems can be categorized broadly as neural network, coupled map lattice, cellular neural network, and cellular automaton [28] (see Table 3.1).

The complexity of solving a problem reduces if each cell has information about the entire system, as for example, in Hopfield networks. On the other hand, with locally connected cells, achieving a global computation through local message passing is more difficult. In locally connected cellular automata, patches of information are passed over large distances and emergent computation is performed. However, the advantage of using locally connected network is the less complex interconnectivity required between cells.
Table 3.1: Cellular computing networks.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Space</th>
<th>Connectivity</th>
<th>Time</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network</td>
<td>discrete</td>
<td>extensive or limited</td>
<td>continuous</td>
<td>continuous</td>
</tr>
<tr>
<td>Coupled Map Lattice</td>
<td>discrete</td>
<td>extensive or limited</td>
<td>discrete</td>
<td>continuous</td>
</tr>
<tr>
<td>Cellular Neural Network</td>
<td>discrete</td>
<td>limited</td>
<td>continuous</td>
<td>continuous</td>
</tr>
<tr>
<td>Cellular Automaton</td>
<td>discrete</td>
<td>limited</td>
<td>discrete</td>
<td>discrete</td>
</tr>
</tbody>
</table>
Chapter 4

DTM

In this chapter, the DTM problem is discussed, prior approaches to solve this problems are reviewed, a new approach is proposed and a mathematical formulation of the problem is derived.

4.1 Motivation

With increased circuit density and clock speed, power dissipation is a problem in today’s high-performance processing units. The quest for increasing overall performance has led to architectural innovations such as multi-threading, multi-core processing, and aggressive execution techniques. However, in the recent years, due to steady miniaturization of transistors and increased chip density, the power densities have increased considerably. Most of the power is consumed in a few localized spots which heat up a good deal faster than the rest of the chip. These hot spots potentially increase leakage currents, cause timing errors and/or result in physical damage, and cause reliability threats to the chip and also impact overall performance. The heating up of chip has become a huge concern since cooling solutions that are expensive are not suitable for consumer products. Also, such cooling techniques are designed to withstand the maximum possible power dissipation, however, the average power dissipation is much lower than the maximum possible value. The disparity between the maximum and typical power dissipation has led to the advocation of the need for DTM techniques in order to ensure that the maximum power levels are not reached. The challenges with respect to reducing the power dissipation and maintaining the chip temperature at working level by not affecting the overall performance have become all the more significant with the advent of multi-processing elements (multi-PeEs) on a single chip.
4.2 DTM: Overview and Strategies

The term DTM refers to a range of hardware and software strategies which work dynamically to control the chip temperature. The packaging and fans of a processing unit are typically designed to be able to allow for normal operation of a chip even at maximum power dissipation. This worst case scenario is less likely to occur than average case power dissipation. DTM techniques target such average case power dissipations and help in lowering on-chip temperature. The important objectives of DTM [31] are:

- to provide inexpensive hardware and/or software responses,
- to reduce power dissipation,
- to impact performance as little as possible.

Figure 4.1 shows how a DTM technique works [31]. The top two lines show the cooling capacity provided without and with DTM techniques. Cooling capability is achieved at lower temperatures with the implementation of DTM techniques. These techniques are triggered and become active after a certain temperature level. The dotted line shows the response of DTM techniques, in other words, the reduced temperature curve while DTM technique is active.

4.2.1 Temperature Trigger Techniques

There are a variety of methods followed in the past to obtain temperature triggers for DTM.

On-Chip Temperature Sensors

Temperature sensors are based on analog CMOS circuits using a current reference whose output current is discretized using a ring oscillator to produce input to a counter [32]. In PowerPC DTM system, thermal feedback relies on an on-chip temperature sensor [33] and Pentium 4 supports an on-die thermal diode whose value can be read using a thermal sensor [34]. Several other approaches are based on feedback from on-chip temperature
Fig. 4.1: Response of DTM technique.

sensors [35–39]. However, the noisy nature of such temperature sensors and the inaccuracy induced to the DTM system due to errors in thermal sensor outputs have been identified [40].

**Dynamic Temperature Model**

Recently, thermal modeling in software has become increasingly popular. Heat transfer within the silicon die is modeled and implemented in software for use in DTM. Predictive application-based thermal model and core-based thermal model have been developed to estimate temperature profile by steady state temperatures and workloads [41].

At an architecture level, HotSpot tool [42] predicts the thermal behavior by modelling the architectural blocks as a circuit of thermal resistances and capacitances (fig. 4.2 [42]). The model can be integrated with a power/performance simulator to determine the thermal stress at a block level.

**On-Chip Activity Counter**

The amount of activity performed by a circuit can be used to estimate its thermal state [43]. This idea is used for the design and use of on-chip activity or performance counter which helps gauge on-chip temperature. On-chip event or performance counter to
augment or replace traditional CMOS temperature sensors have been proposed [44].

**Dynamic and Compile-Time Profiling Analysis**

Performance estimation of applications can be done statically at compile-time where instructions are inserted to specify DTM triggers. Similarly, the profiling of code can be done dynamically to estimate the amount of work performed by an application. There could be certain acceptable level of performance determined before hand, and the DTM system would be triggered for any application exceeding that level [45].

**4.2.2 DTM: Response Techniques**

The main objective of a DTM architecture is to reduce the power and temperature of the computing system and thereby, affecting the performance the least possible. To realize this objective, a number of techniques have been devised to lower power and temperature, which are known as response techniques.
Clock Frequency Scaling

Several processors have multiple clock domains which allow each core or processing element to operate at a different independent frequency. The implementation of such a multiple clock domain requires distribution of an externally generated clock to the phase lock loop (PLL) in each domain. In field-programmable gate arrays (FPGAs), the same is implemented using a digital clock manager which contains a delay-locked loop (DLL) instead of a PLL [46]. Figure 4.3 shows the clocking domains for two of Xilinx Virtex 4 family FPGAs [46].

Clock frequency scaling is effective to reduce power and temperature, however trades a linear performance loss. In addition, changing clock frequencies frequently can incur additional delays, which affects performance.

Voltage and Frequency Scaling

Scaling voltage helps in reducing the dynamic power quadratically, while frequency reduces it linearly. Present day processors [47, 48] have the ability to perform dynamic frequency and voltage scaling which presents tremendous advantage for DTM systems. But with steady miniaturization of circuits, as the future process technologies scale down to lower base supply voltages, dynamic scaling of voltage will be difficult. Since threshold voltages have to be scaled down in order to compensate for the scaling down of operating voltages, there would be a huge rise in the static leakage current as leakage is related to the level of threshold voltage.

Architectural Methods

At the architectural level, approaches like instruction cache (I-cache) toggling, decode throttling and speculation control have been applied to reduce power and temperature. I-cache toggling refers to disabling the instruction fetch unit to feed the pipeline. The disabling can occur for a specific number of cycles for a given amount of time based on the DTM trigger [31]. Certain speculative pipeline gating approaches have been implemented
based on branch confidence, in which the amount of speculation in a pipeline is restricted when a thermal trigger is fed in. The speculation is stalled until a pre-defined number of branches are resolved.

**Task Scheduling and Migration**

Scheduling of hot tasks in cooler regions and cool tasks in hotter regions is another strategy to reduce overall chip temperature. An execution ordering between hot and cold threads for temperature control has been proposed [49]. However, in some cases, it is difficult to determine the run-time behavior of threads. Hence, a method to migrate threads with smaller task size over other threads to reduce the migration cost which comes from task suspension and resumption is presented [50]. Alternately, an offline profiling of applications and run-time clustering of applications to thermal behavior groups for effective task migration by DTM system has been proposed [51]. Also, in FPGA’s, dynamic relocation of tasks is possible with extremely low relocation time with restoration of states by accelerated relocation circuit which can be used for DTM [52].
4.3 DTM: Problem Formulation

In this section, a mathematical formulation of the DTM problem is presented. A multi-
PE system comprised of \( N_{\text{tot}} \) processing elements or computational cores is considered. It is assumed that each PE can operate at a different independent frequency and that task migration from one PE to another possible. \( n \) parallelizable tasks are assumed to be running on \( N_{\text{tot}} \) PEs at any given point in time. Each task \( i \) is assumed to have a priority \( P_i \) associated with it based on the criticality of the task \( i \) in the overall system. The amount of parallelization is denoted by \( N_i \) and frequency of operation by \( F_i \) for any task \( i \). The problem is formulated as a multi-constrained optimization problem where the objective function is

\[
\max f = \int_{t_{\text{on}}}^t \sum_{i=1}^{n} N_i P_i F_i
\]

subject to

\[
\sum_{i=1}^{n} N_i \leq N_{\text{tot}}
\]

\[
F_{\text{min}} \leq F_i \leq F_{\text{max}}
\]

\[
1 \leq N_i \leq N_{\text{tot}} - n + 1
\]

\[
T_i \leq T_{\text{crit}}.
\]

The frequency of operation, operating voltage, number of bit switches, and load capacitance determine the amount of dynamic power consumption by a region on a chip. Temperature of a block in a chip depends on power and the temperature gradient with respect to the vertical and horizontal blocks associated. Performance, for a given task, depends on the frequency of operation and the workload associated. Therefore, it is very critical to make choices of frequency in a system where high performance is targeted while temperature is also a concern. It is proved that the frequency selection problem in this case is an NP hard problem, as it resembles the multiple choice knapsack problem [53].

In the DTM algorithm presented in this thesis, task migration (dynamic change in task parallelism) and dynamic frequency scaling are combined to provide improved performance
numbers, while constraining temperature level.
Chapter 5

Bio-Inspired DTM Technique

In this chapter, a novel DTM technique inspired from stomatal networks is presented. This chapter includes the system of equations used to model the DTM problem using the cellular neural network model and stomatal model, and information regarding the mapping between cellular network outputs and real parameters and algorithm overview.

5.1 DTM Model

As discussed in Chapter 4, two variables, N and F, are varied to manage performance and temperature in a multi-PE computational system. A multi-layer cellular neural network (see sec. 3.2) is used to model these two variables along with temperature.

5.1.1 System of Equations

In the cellular neural network model (fig. 5.1), each cell represents a task. A task has two state variables, N and F, associated with it.

The variable N is a measure of how many units are consumed by a task (in other words, circuit area occupied by the task). Since the total number of resources is limited by a certain area number or number of computational units available, the sum of N values over the entire network need to be constrained by a known number $N_{tot}$. It is assumed that a change in N value would result in increasing or decreasing neighboring tasks’ N values. Hence, a neighborhood interaction is necessary to model the behavior of N. The value $N_i$ for a given task, depends on the priority and temperature value associated with the task. Equation (5.1) shows the differential equation for $N_i$. $i$ represents the location of the cell in the network. $k_t$ is the rate constant based on change in temperature. $N_n$ represents the neighborhood cells (in this case, 4x4 cell matrix). $T_i$ and $P_i$ represent the temperature and
priority for task \( i \). \( a_n \) is the cloning template (3x3 matrix). \( c_n, \delta_n, \) and \( \pi_n \) are the self coefficient and input coefficients, respectively.

\[
\frac{dN_i}{dt} = k_t \left[ -a_n N_i + c_n \sum N_n + (-\delta_n T_i + \pi_n P_i) \right] \tag{5.1}
\]

Equation (5.2) shows the differential equation for \( F \). \( k_t \) is the rate constant based on change in temperature. \( a_f, \delta_f, \) and \( \pi_f \) are the self coefficient and input coefficients for variable \( F \), respectively.

\[
\frac{dF_i}{dt} = k_t \left[ -a_f F_i - \delta_f T_i + \pi_f P_i \right] \tag{5.2}
\]

Equations (5.1) and (5.2) are similar to stomatal eqs. (3.3) and (3.4) in construction.

### 5.1.2 Temperature Modeling

In past, several works have targeted the issue of on-chip temperature modeling. A variety of approaches for modeling the heat transfer in the substrate such as finite element
[54], finite-difference time domain [55], model reduction [56], random walk method [57], and at architectural level, HotSpot tool [42] for block or grid thermal models based on finite element method have been developed.

Any of the above models can be used in combination with the proposed approach. For simplicity, a one-layer first-order RC model is considered. Each cell is assumed to be connected to the ambient or heat sink. Since the vertical heat transfer (between the silicon layer and ambient/heatsink) is far more significant than the horizontal cell-to-cell heat transfer, the connections to the neighboring cells are ignored in this simplified model (shown in fig. 5.2).

The differential equation modeling the heat flow is given by:

\[
\frac{d(T_i - T_{amb})}{dt} = -\lambda[T_i - T_{amb}] + \theta \cdot N_i \cdot F_i \\
\lambda = -\delta \cdot \frac{G_{amb}}{C_{Si \cdot Area}} \\
\theta = \delta \cdot \frac{C_{load \cdot V^2}}{C_{Si \cdot Volume}},
\]

(5.3)

where \( T_i \) is the temperature for region occupied by task \( i \), \( T_{amb} \) is the ambient temperature, \( G_{amb} \) and \( G_{Si} \) are the thermal conductances of ambient and silicon, \( C_{amb} \) and \( C_{Si} \) are the thermal capacitances of ambient and silicon, respectively, \( V \) is the operating voltage and \( \delta \) is the actual time interval between time steps. The results of simulation of the temperature model (eq. (5.3)) for various values of \( N \) and \( F \) is shown in fig. 5.3.

5.1.3 Steady-State Analysis

The cellular neural network attains a steady state when the cloning template is chosen to be symmetric [29]. Here in this model, a neighborhood function of \( N=1 \) is considered (3x3 matrix). An assumption is made that a valid connection exists only between the north, east, south, and west neighbors, thus making the cloning template have five non-zero values. Of the five, four neighbor coefficients (\( \alpha \)) are chosen to be identical and the self-coefficient
Fig. 5.2: Simplified RC model.

Fig. 5.3: Thermal model simulation.
(β) is chosen to be a different value, thus making the cloning template symmetric.

\[
c_n = \begin{pmatrix}
0 & \alpha & 0 \\
\alpha & \beta & \alpha \\
0 & \alpha & 0
\end{pmatrix}
\]

At steady state, an assumption that the state and output values are equal is made. Thus, the eqs. (5.1) and (5.2) reduce to:

**N Steady-State Solution**

\[
0 = -a_n N_i + c_n \sum N_n + (-\delta_n T_i + \pi_n P_i)
\]

\[
N_{tot} = \frac{(-\delta_n \sum T_i + \pi_n \sum P_i)}{a_n - \beta - 4\alpha}
\]

\[
N_{tot} \leq \frac{(-\delta_n n T_{crit} + \pi_n \sum P_i)}{a_n - \beta - 4\alpha}.
\] (5.4)

**F Steady-State Solution**

\[
0 = [-a_f F_i - \delta_f T_i + \pi_f P_i]
\]

\[
[a_f \leq \frac{(-\delta_f n T_{crit} + \pi_f \sum P_i)}{\sum F_i}].
\] (5.5)

The values of \(a_n, c_n, \pi_n, \delta_n, a_f, \pi_f,\) and \(\delta_f\) need to be determined such that the objective function can be maximized.

### 5.2 Mapping Cellular Neural Network and Real-Time System

Based on the cellular neural network equations (eqs. (3.5) - (3.10)), the output values are normalized between -1 and +1. Hence, the output values from the cellular neural network are converted to the physical values using the following equations:
\[ F_{i-phy} = F_{min} + \frac{(F_i + 1)(F_{max} - F_{min})}{2} \]
\[ N_{i-phy} = N_{min} + \frac{(N_i + 1)(N_{max} - N_{min})}{2} \]  \quad (5.6)

The high-level block diagram of the stomatal-based DTM system is shown in fig. 5.4. The cellular neural network outputs are normalized \( N \) and \( F \) values, which are converted to real values and applied to the PE network using dynamic frequency scaling and task migration techniques. The temperature values can be read either by use of thermal sensors or thermal model simulated within the controller and fed back to the cellular neural network.

It is to be noted that changes in value of \( N \) can increase or decrease the area on chip.
operating on a particular task. The controller needs to be programmed to follow a set of rules while changing the value of N for tasks based on cellular neural network’s output.

- The cell neighborhood, as specified by the cellular neural network, need to be maintained at all times.
- The units working on a given task need to be clustered so that communication is possible between parallel units.

An example configuration of tasks on a 64-PE system (each task running in four units initially), modeled by a 4x4 cellular neural network is shown in fig. (5.5).

5.3 Bio-Inspired Algorithm for DTM

The algorithm for DTM is shown in flowchart representation in fig. 5.7.

Fig. 5.5: Example configurations.
The algorithm runs for the time for which the circuit is switched on. The startup temperatures and priorities are used to calculate the initial weights or cloning template. The temperature changes are monitored and for a change in temperature, the cellular neural network is turned on and values of N and F are calculated. The calculated values of N and F are converted to the real values and applied to the circuit by a controller. The temperatures and priorities are polled for changes and weights are re-calculated for any change in priorities.

Figure 5.6 and Table 5.1 illustrate the comparison between the stomatal network and cellular neural network proposed for DTM. The objective problem, dynamics and network structure are compared.

\[
\begin{align*}
\frac{dP_e}{dt} &= k_e \left[(-\delta_e \Delta wa - P_e + \pi_e) - c_e \left(P_e - \{P_e\}\right)\right] \\
\frac{dP_g}{dt} &= k_g \left(-\delta_g \Delta wa - P_g + \pi_g\right) \\
\alpha &= \beta \left[\hat{P}_e - mP_e\right]
\end{align*}
\]

\[
\begin{align*}
\frac{dN_i}{dt} &= k_t \left[-a_n N_i + c_n \sum N_n + (-\delta_n T_i + \pi_n P_i)\right] \\
\frac{dF_i}{dt} &= k_t \left[-a_f F_i - \delta_f T_i + \pi_f P_i\right] \\
\frac{dT_i}{dt} &= -\lambda [T_i - T_{amb}] + \theta F_i N_i
\end{align*}
\]

Fig. 5.6: Stomatal network and cellular neural network.
Fig. 5.7: Bio-inspired DTM algorithm flowchart.
Table 5.1: Stomatal network and cellular neural network for DTM.

<table>
<thead>
<tr>
<th></th>
<th>Stomatal network model</th>
<th>DTM cellular neural network model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization problem</td>
<td>To maximize intake of $CO_2$ for a period of time (typically one day)</td>
<td>To maximize performance of tasks running on chip for a period of time (chip on-time)</td>
</tr>
<tr>
<td>Constraint</td>
<td>water vapor loss</td>
<td>temperature level</td>
</tr>
<tr>
<td>Variables</td>
<td>contribute to both $CO_2$ maximization and water vapor loss</td>
<td>contribute to both performance maximization and temperature increase</td>
</tr>
<tr>
<td>State values</td>
<td>used directly by the system</td>
<td>quantized for use in real system</td>
</tr>
<tr>
<td>Network Connection</td>
<td>With nearest neighbors</td>
<td>With nearest neighbors</td>
</tr>
</tbody>
</table>
Chapter 6

Results

6.1 Weight Determination

As discussed in sec. 5.1.3, a number of coefficients or weights need to be determined in order to configure the network to solve an optimization problem. The weights in this problem are $a_n$, $c_n$, $\pi_n$, $\delta_n$, $a_f$, $\pi_f$, and $\delta_f$. Initially, a mathematic approach to solve the values of weights was used. However, the first derivative was a higher-order polynomial that could not be solved theoretically, and hence a heuristic approach was taken. A search algorithm was implemented to identify the weights for the problem with $n=16$. The search was done for various values of priorities to obtain a global maximum configuration. The graph in fig. 6.1 shows the value of objective function for different values of priorities and the weights in the solution space.

The limits for the weights were chosen in a way that the assumption of state values not exceeding the limits of -1 and +1 was met. Several runs of simulations helped estimate these limits to start the search process. The limits chosen are indication in the fig. 6.1. The simulation setup evaluated the cellular neural network response for a defined time period. The objective function as defined in eq. (4.1) was calculated for each iteration. In the weight search algorithm, the values of the unknown weights were changed by use of nested loops. With a constant value assigned to all other weights, a single weight was allowed to change and assume all values between the limits for each iteration step and the objective function was evaluated and plotted. With the steady state analysis equations (eqs. (5.4) and (5.5)), good choices for certain weights were made which helps the system meet the constraints as defined in eq. (4.2).

A single line in the graph represents the plot of the objective function value for the
entire search run assuming a constant priority input. However, the priority can change during the run-time, and hence to make the choice of weights independent of the priority values, the same search process was repeated for various values of priorities and four of them are shown in this graph.

After obtaining the objective function plot for all possible valid priority inputs, search for maximum values in objective function at each level of priorities was done. The optimum weights are obtained from the intersection of sets of optimum weights (sets based on priority values). This denotes the existence of a basin of attraction (existence of more than one set of weights that can solve the problem). These sets of possible weights are those with which the network can be configured to get desirable results. This helped to ensure that a maximum value in the objective function would be ensured for any change in priorities.

Fig. 6.1: Weights determination.
6.2 Adaptive Response

Using the results of weights, simulations were run on a 16 node (4x4) network for 64-PE system. Simulations were run for various priority configurations. A snapshot of the simulation results is shown in fig. 6.2.

The results indicate adaptive response of the system for variations in the values of task priorities and temperature. The lighter shades correspond to lower values and darker shades higher values of temperature, frequency and N. Figure 6.2(a) is the initial setup where temperature is low, frequency maximum and task parallelism level equal to start with. With the inputs of task priority shown in fig. 6.2(b), the N values change, resulting in change in temperature as shown in fig. 6.2(b). The configuration of N and F values in fig. 6.2(b) results in different values of temperature at different blocks. This results in changes in F and N to maintain the temperature at working level, as shown in fig. 6.2(c) and fig. 6.2(d). In fig. 6.2(e), a new set of priorities is applied, which in turn, changes the value of N and F; the subsequent change in temperature level is shown. The change in temperature further results in changes in N and F, as illustrated in fig. 6.2(f), resulting in a uniform temperature profile. It can be seen that the network tries to avoid hot spot conditions through changes in N and F values and at the same time takes care of priorities of tasks.

The numerical values for the inputs and outputs are shown in the fig. 6.3.

6.3 Thermal Profile

Figures 6.4 and 6.5 illustrate the temperature variation during the simulation. Figure 6.4 shows the temperature of various tasks. It can be noted that the network keeps the temperature under the threshold value (80°C in this setup) for any sudden increase in temperature. Figure 6.5 shows that if tasks have different starting temperatures to begin with, they would all converge to a steady state value by means of change in values of N and F.
Fig. 6.2: Adaptive response.
<table>
<thead>
<tr>
<th>Input: Priority</th>
<th>Input: Temperature</th>
<th>Output: N values</th>
<th>Output: F values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01 0.02 0.03 0.04</td>
<td>47.65 47.76 47.88 47.95</td>
<td>2 2 2 2</td>
<td>350 350 350 350</td>
</tr>
<tr>
<td>0.05 0.01 0.02 0.03</td>
<td>48.02 47.57 47.76 47.88</td>
<td>2 2 2 2</td>
<td>350 350 350 350</td>
</tr>
<tr>
<td>0.04 0.05 0.95 0.95</td>
<td>48.08 48.19 58.52 58.52</td>
<td>2 2 7 7</td>
<td>350 350 350 350</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>59.1 59.1 59.18 59.18</td>
<td>7 7 7 7</td>
<td>350 350 350 350</td>
</tr>
</tbody>
</table>

| 0.01 0.02 0.03 0.04 | 52.51 52.66 52.85 53.02 | 1 1 1 1 | 344 345 344 344 |
| 0.05 0.01 0.02 0.03 | 52.84 52.17 52.05 52.81 | 1 1 1 1 | 350 350 345 345 |
| 0.04 0.05 0.95 0.95 | 51.36 53.51 69.02 69.02 | 1 1 8 8 | 340 340 345 345 |
| 1 1 1 1 | 69.89 69.9 70.21 70.22 | 9 9 10 10 | 345 345 343 343 |

| 0.01 0.02 0.03 0.04 | 54.63 54.65 54.76 54.8 | 3 3 4 4 | 300 301 300 300 |
| 0.05 0.01 0.02 0.03 | 54.49 54.31 54.94 54.7 | 3 3 3 3 | 306 306 301 301 |
| 0.04 0.05 0.95 0.95 | 55.09 55.12 58.49 58.49 | 4 4 5 5 | 296 296 301 301 |
| 1 1 1 1 | 58.66 58.66 58.3 58.81 | 5 5 5 5 | 301 301 299 299 |

| 1 1 1 1 | 70.67 70.7 70.7 71.16 | 6 6 6 6 | 350 350 350 350 |
| 0.01 0.01 0.01 0.01 | 57.82 58.25 57.67 57.67 | 3 3 3 3 | 306 306 301 301 |
| 0.01 0.05 0.01 0.01 | 58.02 68.91 59.88 59.88 | 3 5 2 2 | 293 348 292 292 |
| 0.01 0.01 0.01 0.95 | 60.37 60.37 59.66 71.27 | 4 4 3 5 | 289 289 247 350 |

| 1 1 1 1 | 61.81 61.81 61.81 61.79 | 5 5 5 5 | 340 340 340 339 |
| 0.01 0.01 0.01 0.01 | 57.95 57.96 57.71 57.72 | 4 4 4 3 | 240 232 225 226 |
| 0.01 0.05 0.01 0.01 | 57.39 61.7 57.32 57.32 | 3 4 3 3 | 204 341 201 201 |
| 0.01 0.01 0.01 0.95 | 57.39 57.39 56.51 61.67 | 4 4 4 4 | 189 189 101 338 |

Fig. 6.3: Numerical results.
Fig. 6.4: Temperature simulation.

Fig. 6.5: Steady-state convergence of temperature.
6.4 Advantages

The bio-inspired technique for DTM has several advantages over the other existing techniques.

- This technique is based on parallel cells working in unison to converge to an optimal solution, and hence is extremely parallel and faster than any sequential approach.
- The network is scalable and can be customized to extend to any number of tasks.
- The cells in the network are exact replicas of a basic cell node and have very less interconnecting wires (4-cell neighborhood). Both these factors make it easily less complex for realizing on a chip fabric.
Chapter 7
Conclusions and Future Work

In this thesis, stomatal network dynamics and its application to solve constrained optimization problem is discussed. This work proposes a novel DTM technique to maximize performance and constrain temperature. This technique uses cellular neural network to model tasks running in processing elements. The outputs of the network are frequency and amount of parallelism of tasks. The proposed method is simulated for a 64-processing element system with 16 tasks running on it. The simulation results show that the network adapts itself based on the inputs, priority of tasks and temperature values. The weights of the network are chosen in such a way that the performance value is maximized and temperature is constrained.

The main challenge in this thesis is to understand how the stomatal network works and relate other constrained optimization problems to the problem solved by stomata. Modeling of the DTM problem is done with a knowledge of the optimization problem, stomatal network using cellular neural network. The next challenge is the configuration of the network such that it solves the optimization problem with all constraints satisfied. This is done by a mathematical formulation of the problem and running searches in the solution space for finding the basin of attraction of weights for optimal solution. A simplified temperature model is chosen but the solution can work with any temperature model or sensor-based temperature triggers as well.

Additional improvements and related work that can be build on this thesis are as follows.

- DTM technique can be experimentally verified on-chip using an FPGA’s in-built digital clock manager and partial relocation methods for implementing change in frequency and parallelism, respectively.
• The technique can be easily extended to a three-dimensional chip for DTM. This would require changes in the temperature modeling and the network interconnections. A three-dimensional cellular neural network can be used to model tasks running on a three-dimensional chip.

• The variables in this model are frequency and parallelism. However, there could be other variables that can be included to form additional layers that would help improving the problem solution quality, e.g., voltage, time to relocate, task deadlines, etc.

• The performance is assumed to be a weighted product of frequency and number of parallel units combined with task priority. However, not all the work done by parallel units could be useful (memory reads or writes, communication latency, idle times) which could result in a more complex objective function. Use of a more accurate formulation of performance can yield better solutions.

• As discussed earlier, a more accurate temperature model can be applied which includes lateral heat transfer laterally, effect of copper, heat sink, and packaging layers on the chip.

• Identification of other optimization problems that can be related to the constrained optimization problem solved by stomtal network and application of the stomatal rules to solve them for broader application of the stomatal model.
References


[53] J. Sartori and R. Kumar, “Proactive peak power management for many-core architectures.”

