THE UBIQUITOUS SEARCH
(METHODS TO ESCAPE FROM LOCAL MINIMA)

Benjamin W. Wah

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, IL 61801, USA
b-wah@uiuc.edu
URL: http://manip.crhc.uiuc.edu

November 1996
The Ubiquitous Search

Outline

- Characteristics of
  - Search Problems
  - Search Algorithms
- Existing methods to
  - Help escape from local minima
  - Handle constraints
- NOVEL: Nonlinear Optimization With External Lead
- Applications of NOVEL
  - Nonlinear continuous constrained optimization problems
  - Filter bank design problems
  - Nonlinear discrete satisfiability problems
  - Feedforward neural network learning problems
Motivations

- Many real-world applications
  - Artificial intelligence
  - Logic
  - Computer aided design
  - Database query processing
  - Planning
  - Scheduling

- Complete methods cannot handle large problems

- Global search versus local search
Characteristics of Search Problems

- Levels of search problem
  - Problem instance level
  - Meta level: generalization of solution

- Search space
  - Finite/infinite

- Variables
  - Fixed and well defined/undefined (and possibly unbounded)
  - Discrete/continuous/mixed/symbolic
Characteristics of Search Problems (cont’d)

- Objective
  - Well defined/undefined
  - Linear/nonlinear/symbolic

- Objective measures
  - Deterministic/probabilistic
  - Resource measures

- Constraints
  - Hard/soft constraints
  - Linear/nonlinear/symbolic
  - Resource constraints
Characteristics of Search Algorithms

• Representation of search space
  – Search complexity
• Decomposition strategies
• Heuristic predictor or direction finder
  – Relaxation algorithms
• Mechanisms to help escape from local minima
• Mechanisms to handle constraints
• Stopping conditions
• Resource scheduling strategies
METHODS TO HELP ESCAPE FROM LOCAL MINIMA
Local Minima

- Tall hills that are difficult to overcome
- Gradients vary by many orders of magnitude
- Shallow basins with small slopes
- Deep valleys with steep slopes
The Ubiquitous Search

Existing Methods to Help Escape from Local Minima

- Nonlinear minimization methods
  - Local minimization methods
  - Global minimization methods
    - Deterministic methods
      - Covering methods
      - Generalized descent methods (gradient-based)
        - NOVEL method
      - Trajectory & penalty methods
    - Probabilistic methods
      - Clustering methods
      - Random search methods
      - Methods based on stochastic models

Benjamin W. Wah
Existing Methods (cont’d)

Deterministic methods
- Covering – Detect regions not containing global minima and exclude them
- Trajectory – Modify differential equations modeling local descents
- Penalty – Modify objective function to avoid redetermination of the same local minima

Probabilistic methods
- Clustering – Group points around local minima (difficult when terrain is rugged)
- Random – Single start, multi-start, random line search, adaptive random search, evolutionary algorithms, simulated annealing
- Stochastic – Use random variables to model unknown values of objective (Bayesian)
Existing Methods: Summary

- Covering methods and methods based on stochastic models are inefficient in dealing with problems with more than 20 variables.

- Generalized descent methods and clustering methods are inefficient in dealing with problems with many local minima.
  - Descent methods get trapped in local minima.

- Random search methods are inefficient due to randomness and redetermination of local minima.
HANDLING CONSTRAINTS
Existing Methods for Handling Constraints

- Non-transformational approaches
  - Discarding methods
  - Back-to-feasible-region methods

- Transformational approaches
  - Penalty methods
    * Optimize sum of objective and constraints weighted by penalties
    * Penalize suboptimal solutions weighted by penalties in objective
  - Barrier methods: add new barriers during search
  - Lagrange-multiplier methods
Lagrangian Methods

- Optimization problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad h(x) = 0
\end{align*}
\]

- Lagrangian/Augmented Lagrangian functions

\[
\begin{align*}
L(x, \lambda) &= f(x) + \lambda^T h(x) \\
\mathcal{L}(x, \lambda) &= f(x) + \|h(x)\|_2^2 + \lambda^T h(x)
\end{align*}
\]

- Sufficient conditions for optimality: System of differential equations

\[
\nabla_x L(x, \lambda) = 0 \\
\nabla_\lambda L(x, \lambda) = 0
\]
Lagrangian Methods (cont’d)

- Two counteracting forces to converge to saddle points
  - Gradient descent in $x$ space ($\frac{dx}{dt} = -\nabla_x \mathcal{L}(x, \lambda)$)
    * When constraints are violated: minimize violation
    * When constraints are not violated: minimize objective
      ($\lambda$ carries no weight)
  - Gradient ascent in $\lambda$ space ($\frac{d\lambda}{dt} = \nabla_\lambda \mathcal{L}(x, \lambda)$)
    * When constraints are violated, increase $\lambda$ to increase weight of violation

- More effective than penalty methods in adjusting $\lambda$

- Handling inequality constraints
  - Slack variable method
  - MaxQ method
Discrete Lagrangian Methods

- Discrete optimization problem

  \[
  \text{minimize} \quad f(x) \\
  \text{subject to} \quad h(x) = 0, \quad x \in Z
  \]

- Discrete Lagrangian function: \( L(x, \lambda) = f(x) + \lambda^T h(x) \)

- Dynamic system

  \[
  x_{k+1} = x_k - \Delta x L(x_k, \lambda_k) \\
  \lambda_{k+1} = \lambda_k + h(x_k)
  \]

- Discrete Saddle-Point Theorem: \( F(x^*, \lambda) \leq F(x^*, \lambda^*) \leq F(x, \lambda^*) \)

- Fixed Point Theorem: Feasible solution is reached if dynamic system terminates
NOVEL: NONLINEAR OPTIMIZATION VIA EXTERNAL LEAD
Features of NOVEL

• Global search: locating promising regions
  – A user-defined trace function leading the search
  – Local minima attracting the search trajectory

• Local search
  – Gradient descent
  – Lagrangian search
A Simple Example

- Minimizing Levy’s No. 3 function of two variables

\[ f_{i3}(x) = \sum_{i=1}^{5} i \cos[(i - 1)x_1 + i] \sum_{j=1}^{5} j \cos[(j + 1)x_2 + j] \]
The Ubiquitous Search
The Ubiquitous Search

Framework of *NOVEL*

- Global search phase
  - Three stages in tandem to explore search space
  - Locate promising regions with good local minima

- Local search phase
  - Descent methods, e.g. gradient descent
  - Conjugate gradient
  - Quasi-Newton’s method

\[
T(t) \rightarrow \begin{cases} 
\dot{X}_1(t) = P(\nabla_X f(X_1(t))) + Q(T(t), X_1(t)) 
\end{cases} 
\]

Global Search Phase

\[
X_1(t) \rightarrow \begin{cases} 
\dot{X}_2(t) = P(\nabla_X f(X_2(t))) + Q(X_1(t), X_2(t)) 
\end{cases} 
\]

Descent Methods
Select starting points from \(X_1(t), X_2(t), X_3(t)\);
Apply gradient descent from these points

Local Search Phase

\[
X_2(t) \rightarrow \begin{cases} 
\dot{X}_3(t) = P(\nabla_X f(X_3(t))) + Q(X_2(t), X_3(t)) 
\end{cases} 
\]

Local Minima
Illustration of Global Search Phase

Trace

Trace direction

Trajectory

Moving direction

Gradient direction

Trace function

Trajectory 1

Trajectory 2

Trajectory 3
The Ubiquitous Search

Uniform Traversal of Search Space by $T(t)$

- For each dimension, search the whole space from coarse to fine

- $T(t)$ — Aperiodic trace function searching from coarse to fine
  
  $= \rho \sin \left[ 2\pi \left( \frac{t}{2} \right)^{0.95} - \frac{0.45(i-1)}{n} \right] + \frac{2\pi(i-1)}{n}$

  - $t$: autonomous variable
  - $n$: number of dimensions
  - $i$: $i$’th dimension
  - $\rho$: search range
Mathematical Formulation of Global Search Phase

- Generic formulation to specify a trajectory through variable space $X$.

$$
\frac{dX(t)}{dt} = P(\nabla_X f(X(t))) + Q(T(t), X(t))
$$

- $f(X)$: Error function to be minimized
- $\nabla_X f(X)$: Gradient of $f(X)$
- $P(\nabla_X f(X(t)))$ enables gradient to attract the trajectory
- $Q(T(t), X(t))$ allows trace function $T(t)$ to lead the trajectory

- One simple trajectory through variable space $X$

$$
\frac{dX(t)}{dt} = -\mu_g \nabla_X f(X(t)) - \mu_t (X(t) - T(t))
$$

- $\mu_g$ and $\mu_t$ are positive real numbers
The Ubiquitous Search

**Insufficient Lateral Force**

Downward force due to gradient

$>>$ Lateral force due to trace

-1000

5
Lateral force increased significantly by scaling variables
Solution Methods

- Differential equation solver, e.g. LSODE package for solving ordinary differential equations from netlib
  - Slow
  - Accurate

- Finite difference equation solver
  \[
  X(t + \delta t) = X(t) + \delta t \{-\mu_g \nabla f(X) - \mu_t [X(t) - T(t)]\}
  \]
  - Fast
  - Approximate
APPLICATION 1: NONLINEAR CONTINUOUS CONSTRAINED OPTIMIZATION
Lagrangian Search and Trace

• Dynamic system

\[
\frac{dx}{dt} = -\nabla_x \mathcal{L}(x(t), \lambda(t)) + w \star [\mathcal{L}_x(x(t), \lambda(t)) - T(x(t))] \\
\frac{d\lambda}{dt} = \nabla_\lambda \mathcal{L}(x(t), \lambda(t))
\]

• Lyapunov function as stopping condition

\[ F(x, \lambda) = || - \nabla_x \mathcal{L}(x, \lambda) ||^2 + || \nabla_\lambda \mathcal{L}(x, \lambda) ||^2. \]

• Handling inequality constraint \( g(x) \leq 0 \)

\[
\left[ max^2(0, \mu_i + g_i(X)) - \mu_i^2 \right] = 0 \\
\left[ \mu_i max^{q_i}(0, g_i(X)) \right] = 0
\]
Example: Levy’s No. 3 Function & Elliptic Constraint

\[ x-y \text{ plot} \]

\[ x-\lambda \text{ plot} \]
Example (cont’d)

Objective, constraint, & trace

Objective, constraint, & trajectory
### The Ubiquitous Search

<table>
<thead>
<tr>
<th>Problem ID</th>
<th>Search Range</th>
<th>NOVEL Search Time Limit</th>
<th>Best Known Solutions</th>
<th>Epperly’s Slack w/o Scaling Solutions</th>
<th>Slack w/ Scaling Solutions</th>
<th>MaxQ Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1</td>
<td>1.0</td>
<td>3279</td>
<td>-17.00</td>
<td>-17.00</td>
<td>-</td>
<td>-17.00</td>
</tr>
<tr>
<td>2.2.1</td>
<td>10.0</td>
<td>5856</td>
<td>-213.00</td>
<td>-213.00</td>
<td>-</td>
<td>-213.00</td>
</tr>
<tr>
<td>2.3.1</td>
<td>10.0</td>
<td>57404</td>
<td>-15.00</td>
<td>-15.00</td>
<td>-</td>
<td>-15.00</td>
</tr>
<tr>
<td>2.4.1</td>
<td>10.0</td>
<td>29829</td>
<td>-11.00</td>
<td>-11.00</td>
<td>-</td>
<td>-11.00</td>
</tr>
<tr>
<td>2.5.1</td>
<td>1.0</td>
<td>2937</td>
<td>-268.00</td>
<td>-268.00</td>
<td>-</td>
<td>-268.00</td>
</tr>
<tr>
<td>2.6.1</td>
<td>1.0</td>
<td>3608</td>
<td>-39.00</td>
<td>-39.00</td>
<td>-</td>
<td>-39.00</td>
</tr>
<tr>
<td>2.7.1(1)</td>
<td>40.0</td>
<td>68563</td>
<td>-394.75</td>
<td>-394.75</td>
<td>-</td>
<td>-394.75</td>
</tr>
<tr>
<td>2.7.1(2)</td>
<td>40.0</td>
<td>51175</td>
<td>-884.75</td>
<td>-884.75</td>
<td>-</td>
<td>-884.75</td>
</tr>
<tr>
<td>2.7.1(3)</td>
<td>40.0</td>
<td>170751</td>
<td>-8695.00</td>
<td>-8695.00</td>
<td>-</td>
<td>-8695.00</td>
</tr>
<tr>
<td>2.7.1(4)</td>
<td>40.0</td>
<td>203</td>
<td>-754.75</td>
<td>-754.75</td>
<td>-</td>
<td>-754.75</td>
</tr>
<tr>
<td>2.7.1(5)</td>
<td>40.0</td>
<td>97470</td>
<td>-4150.40</td>
<td>-4150.40</td>
<td>-</td>
<td>-4150.40</td>
</tr>
<tr>
<td>2.8.1</td>
<td>25.0</td>
<td>158310</td>
<td>15990.00</td>
<td>15990.00</td>
<td>15639.00</td>
<td>15639.00</td>
</tr>
<tr>
<td>3.1.1</td>
<td>5000.0</td>
<td>352305</td>
<td>7049.25</td>
<td>-</td>
<td>-</td>
<td>7049.25</td>
</tr>
<tr>
<td>3.2.1</td>
<td>50.0</td>
<td>47346</td>
<td>-30665.50</td>
<td>-30665.50</td>
<td>-</td>
<td>-30665.50</td>
</tr>
<tr>
<td>3.3.1</td>
<td>10.0</td>
<td>803</td>
<td>-310.00</td>
<td>-310.00</td>
<td>-</td>
<td>-310.00</td>
</tr>
<tr>
<td>3.4.1</td>
<td>5.0</td>
<td>199</td>
<td>-4.00</td>
<td>-4.00</td>
<td>-</td>
<td>-4.00</td>
</tr>
<tr>
<td>4.3.1</td>
<td>5.0</td>
<td>20890</td>
<td>-4.51</td>
<td>-4.51</td>
<td>-</td>
<td>-4.51</td>
</tr>
<tr>
<td>4.4.1</td>
<td>5.0</td>
<td>73</td>
<td>-2.217</td>
<td>-2.217</td>
<td>-</td>
<td>-2.217</td>
</tr>
<tr>
<td>4.5.1</td>
<td>5.0</td>
<td>16372</td>
<td>-11.96</td>
<td>-13.40</td>
<td>-</td>
<td>-13.40</td>
</tr>
<tr>
<td>4.6.1</td>
<td>5.0</td>
<td>4435</td>
<td>-5.51</td>
<td>-5.51</td>
<td>-</td>
<td>-5.51</td>
</tr>
<tr>
<td>4.7.1</td>
<td>5.0</td>
<td>423</td>
<td>-16.74</td>
<td>-16.74</td>
<td>-</td>
<td>-16.75</td>
</tr>
<tr>
<td>5.2.1</td>
<td>50.0</td>
<td>240829</td>
<td>1.567</td>
<td>-</td>
<td>1.567</td>
<td>1.567</td>
</tr>
<tr>
<td>5.4.1</td>
<td>50.0</td>
<td>374850</td>
<td>1.86</td>
<td>-</td>
<td>1.86</td>
<td>1.86</td>
</tr>
<tr>
<td>6.2.1</td>
<td>100.0</td>
<td>3017</td>
<td>400.00</td>
<td>400.00</td>
<td>400.00</td>
<td>400.00</td>
</tr>
<tr>
<td>6.3.1</td>
<td>100.0</td>
<td>2756</td>
<td>600.00</td>
<td>600.00</td>
<td>600.00</td>
<td>600.00</td>
</tr>
<tr>
<td>6.4.1</td>
<td>100.0</td>
<td>3340</td>
<td>750.00</td>
<td>750.00</td>
<td>750.00</td>
<td>750.00</td>
</tr>
<tr>
<td>7.2.1</td>
<td>100.0</td>
<td>162643</td>
<td>56825.00</td>
<td>56825.00</td>
<td>56825.00</td>
<td>56825.00</td>
</tr>
<tr>
<td>7.3.1</td>
<td>150.0</td>
<td>228320</td>
<td>46266.00</td>
<td>-</td>
<td>46266.00</td>
<td>44903.00</td>
</tr>
<tr>
<td>7.4.1</td>
<td>150.0</td>
<td>631029</td>
<td>35920.00</td>
<td>-</td>
<td>35920.00</td>
<td>35920.00</td>
</tr>
</tbody>
</table>

Benjamin W. Wah
APPLICATION 2: QMF FILTER-BANK DESIGN
# QMF Filter Bank Design

<table>
<thead>
<tr>
<th>Filter</th>
<th>Design Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Filter</td>
<td>Min amplitude distortion</td>
</tr>
<tr>
<td>Bank</td>
<td>Min aliasing distortion</td>
</tr>
<tr>
<td>Single Filter</td>
<td>Min phase distortion</td>
</tr>
<tr>
<td></td>
<td>Min stopband ripple ($\delta_s$)</td>
</tr>
<tr>
<td></td>
<td>Min passband ripple ($\delta_p$)</td>
</tr>
<tr>
<td></td>
<td>Min transition band error ($E_t$)</td>
</tr>
<tr>
<td></td>
<td>Min stopband energy ($E_s$)</td>
</tr>
<tr>
<td></td>
<td>Max passband flatness ($E_p$)</td>
</tr>
</tbody>
</table>

\[ |H(e^{j\omega})| \]

![Graph showing frequency response of filters]

\[ E_p \quad E_s \quad \omega_p \quad \omega_s \quad 0 \quad \pi \]
Example: 24D QMF Design

Passband frequency response

Reconstruction error
Initial Results: Better Designs
<table>
<thead>
<tr>
<th>Filter-type</th>
<th>$E_r$</th>
<th>$\delta_p$</th>
<th>$E_p$</th>
<th>$\delta_s$</th>
<th>$E_s$</th>
<th>$T_R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16a</td>
<td>0.986</td>
<td>1.000</td>
<td>0.858</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>16b</td>
<td>0.985</td>
<td>1.000</td>
<td>0.894</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>16c</td>
<td>0.820</td>
<td>1.000</td>
<td>0.926</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>24b</td>
<td>0.964</td>
<td>1.000</td>
<td>0.802</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>24c</td>
<td>0.893</td>
<td>0.977</td>
<td>0.588</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>24d</td>
<td>0.753</td>
<td>1.000</td>
<td>0.768</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>32c</td>
<td>0.959</td>
<td>1.000</td>
<td>0.748</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>32d</td>
<td>0.870</td>
<td>1.000</td>
<td>0.802</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>32e</td>
<td>0.712</td>
<td>1.000</td>
<td>0.896</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>48c</td>
<td>0.787</td>
<td>0.972</td>
<td>0.793</td>
<td>0.970</td>
<td>0.999</td>
<td>1.000</td>
</tr>
<tr>
<td>48d</td>
<td>0.947</td>
<td>1.000</td>
<td>0.756</td>
<td>0.999</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>48e</td>
<td>0.852</td>
<td>1.000</td>
<td>0.842</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>64d</td>
<td>0.821</td>
<td>0.981</td>
<td>0.767</td>
<td>0.963</td>
<td>0.999</td>
<td>1.000</td>
</tr>
<tr>
<td>64e</td>
<td>0.843</td>
<td>1.000</td>
<td>0.749</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Initial Results: Different Constraints

![Graph showing the relationship between constraint ratio and solution ratio for different constraints. The graph compares 24D and 32D scenarios.](image-url)
APPLICATION 3: DISCRETE SATISFIABILITY
Satisfiability (SAT) Problem

- Given
  - a set of $m$ clauses $C_1, C_2, \cdots, C_m$ on $n$ variables
    \[ X = (x_1, x_2, \cdots, x_n) \quad x_i \in \{0, 1\} \]
  - Boolean formula in conjunctive normal form (CNF)
    \[ C_1 \cap C_2 \cap \cdots \cap C_m \]

- Find a truth assignment or derive infeasibility
**Alternative Formulations**

- Discrete constrained decision problem without objective

Find $x$ such that $U_i(X) = 0$ \ for \ $i = 1, \ldots, m$

where $U_i(x) = \begin{cases} 0 & \text{assignment } x \text{ satisfies } C_i, \\ 1 & \text{otherwise}. \end{cases}$

- Examples: Resolution, backtracking, constraint satisfaction, Davis-Putnam’s algorithm

- High computational complexity
Alternative Formulations (cont’d)

• Discrete unconstrained formulation

\[
\min N(x) = \sum_{i=1}^{m} U_i(x)
\]

– Local search methods
  * GSAT
  * WSAT
  * Gu’s methods
  * Simulated annealing
  * Genetic algorithm
– Can solve many large SAT problems efficiently
– May not work well when there are very few local minima
– Restarts may bring the search to a completely new terrain
Alternative Formulations (cont’d)

- Continuous unconstrained formulation

\[ c_i(x) = \prod_{j=1}^{m} a_{i,j}(x_j) \]

\[ a_{i,j}(x_j) = \begin{cases} 
(1 - x_j)^2 & \text{if } x_j \text{ in } C_i \\
 x_j^2 & \text{if } x_j \text{ in } C_i \\
 1 & \text{otherwise}
\end{cases} \]

Objective: \( \min \sum_i c_i(x) \)

- Gu’s UniSAT model
- Local search methods: gradient descent, conjugate gradient, Quasi-Newton
- Computationally expensive
Alternative Formulations (cont’d)

• Continuous constrained formulation

\[
\min_{x \in E^m} f(x) = \sum_{i=1}^{n} c_i(x)
\]

subject to \( c_i(x) = 0 \quad \forall i \in \{1, 2, \ldots, n\} \)

− Smooth out local minima in discrete space

− Lagrangian methods

− Very expensive
Our SAT Formulation

- Constrained optimization with artificial (discrete) objective

\[ \min \quad N(X) = \sum_{i=1}^{m} U_i(X) \]

subject to \( U_i(X) = 0 \quad i = 1, \ldots, m \)

- Local minima satisfying constraints are also global minima

- Use objective to guide search

- Use constraints to bring search out of local minima without restarts
Gradient Operator $\Delta_X L(X, \lambda)$

$L = 10$

$N(A) = \{B, C, D, E\}$
The Ubiquitous Search

An Implementation to solve SAT Problems

Set initial point \( x \)
\[ \textbf{while } x \text{ is not a solution, i.e., } N(x) > 0 \]
\[ \textbf{while } \Delta_x L(x, \lambda) \neq 0 \]
\[ \text{update } x: x \leftarrow x - \Delta_x L(x, \lambda) \]
\[ \textbf{end while} \]
\[ \text{update } \lambda: \lambda \leftarrow \lambda + c \times U(x) \]
\[ \textbf{end while} \]
The Ubiquitous Search

Comparing DLM with GSAT, WSAT, DP, IP and SA

<table>
<thead>
<tr>
<th>Problem Id</th>
<th>No. of Var.</th>
<th>No. of Clauses</th>
<th>DLM Version 2</th>
<th>WSAT</th>
<th>GSAT</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SS 10/51</td>
<td>Challenge</td>
<td># Iter.</td>
<td></td>
</tr>
<tr>
<td>ssa7552-038</td>
<td>1501</td>
<td>3575</td>
<td>0.228</td>
<td>0.3</td>
<td>7970</td>
<td>2.3</td>
</tr>
<tr>
<td>ssa7552-158</td>
<td>1363</td>
<td>3034</td>
<td>0.088</td>
<td>0.1</td>
<td>2169</td>
<td>2</td>
</tr>
<tr>
<td>ssa7552-159</td>
<td>1363</td>
<td>3032</td>
<td>0.085</td>
<td>0.1</td>
<td>2154</td>
<td>0.8</td>
</tr>
<tr>
<td>ssa7552-160</td>
<td>1391</td>
<td>3126</td>
<td>0.097</td>
<td>0.1</td>
<td>3116</td>
<td>1.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem Id.</th>
<th>No. of Var.</th>
<th>No. of Clauses</th>
<th>DLM Version 2</th>
<th>GSAT</th>
<th>Integer Prog.</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SS 10/51</td>
<td>Challenge</td>
<td># Iter.</td>
<td></td>
</tr>
<tr>
<td>ii16a1</td>
<td>1650</td>
<td>19368</td>
<td>0.122</td>
<td>0.128</td>
<td>819</td>
<td>2</td>
</tr>
<tr>
<td>ii16b1</td>
<td>1728</td>
<td>24792</td>
<td>0.265</td>
<td>0.310</td>
<td>1546</td>
<td>12</td>
</tr>
<tr>
<td>ii16c1</td>
<td>1580</td>
<td>16467</td>
<td>0.163</td>
<td>0.173</td>
<td>797</td>
<td>1</td>
</tr>
<tr>
<td>ii16d1</td>
<td>1230</td>
<td>15901</td>
<td>0.188</td>
<td>0.233</td>
<td>908</td>
<td>3</td>
</tr>
<tr>
<td>ii16e1</td>
<td>1245</td>
<td>14766</td>
<td>0.297</td>
<td>0.302</td>
<td>861</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem Identification</th>
<th>No. of Var.</th>
<th>No. of Clauses</th>
<th>DLM Version 3</th>
<th>GSAT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Time</td>
<td>Success</td>
</tr>
<tr>
<td>g125.17</td>
<td>2125</td>
<td>66272</td>
<td>1390.32</td>
<td>10/10</td>
</tr>
<tr>
<td>g125.18</td>
<td>2250</td>
<td>70163</td>
<td>3.197</td>
<td>10/10</td>
</tr>
<tr>
<td>g250.15</td>
<td>3750</td>
<td>233965</td>
<td>2.798</td>
<td>10/10</td>
</tr>
<tr>
<td>g250.29</td>
<td>7250</td>
<td>454622</td>
<td>1219.56</td>
<td>9/10</td>
</tr>
</tbody>
</table>
### Results On Difficult But Satisfiable DIMACS Benchmarks

<table>
<thead>
<tr>
<th>Prob. Id.</th>
<th>Succ. Ratio</th>
<th>Time in CPU seconds</th>
<th>Prob. Id.</th>
<th>Succ. Ratio</th>
<th>Time in CPU seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>par8-1</td>
<td>10/10</td>
<td>4.780</td>
<td>0.133</td>
<td>14.383</td>
<td>par16-1-c</td>
</tr>
<tr>
<td>par8-2</td>
<td>10/10</td>
<td>5.058</td>
<td>0.100</td>
<td>13.067</td>
<td>par16-2-c</td>
</tr>
<tr>
<td>par8-3</td>
<td>10/10</td>
<td>9.903</td>
<td>0.350</td>
<td>21.150</td>
<td>par16-3-c</td>
</tr>
<tr>
<td>par8-4</td>
<td>10/10</td>
<td>5.842</td>
<td>0.850</td>
<td>16.433</td>
<td>par16-4-c</td>
</tr>
<tr>
<td>par8-5</td>
<td>10/10</td>
<td>14.628</td>
<td>1.167</td>
<td>34.900</td>
<td>par16-5-c</td>
</tr>
<tr>
<td>hanoi4</td>
<td>1/10</td>
<td>682.6</td>
<td>682.6</td>
<td>682.6</td>
<td>f1000</td>
</tr>
<tr>
<td>f600</td>
<td>10/10</td>
<td>16.9</td>
<td>2.1</td>
<td>37.2</td>
<td>f2000</td>
</tr>
</tbody>
</table>

- Still cannot solve 16 satisfiable DIMACS benchmark problems
  - \textit{par}16-1 thru \textit{par}16-5
  - \textit{par}32-1 thru \textit{par}32-5
  - \textit{par}32-1-c thru \textit{par}32-5-c
  - \textit{hanoi}5
APPLICATION 4: FEEDFORWARD NEURAL-NETWORK LEARNING
Two-spiral problem

- Discriminate between two sets of points that lie on two distinct spirals in the $x$-$y$ plane
- Best known network: 9 hidden units with 75 weights
- Training and test data set

194 training patterns

194 testing patterns
The Ubiquitous Search

2-D Classification Graphs for 3, 4, 5, 6 Hidden Units

**NOVEL:** 17 wt.  
25 wt.  
33 wt.  
42 wt.

Simulated Annealing

Best 4 hidden unit network: 77.5 hours – 99% correct, 89.4 hours – 100%
### Experimental Results

<table>
<thead>
<tr>
<th>Problems</th>
<th># of H.U.</th>
<th># of Wts.</th>
<th><strong>TN-MS</strong></th>
<th><strong>NOVEL</strong></th>
<th><strong>TN-MS + NOVEL</strong></th>
<th>CPU time limits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Correct %</td>
<td>Correct %</td>
<td>Correct %</td>
<td># time units</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>training</td>
<td>test</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td># of</td>
<td></td>
<td># of</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>restarts</td>
<td></td>
<td>restarts</td>
<td></td>
</tr>
<tr>
<td>Sonar</td>
<td>2</td>
<td>125</td>
<td>98.1</td>
<td>90.4</td>
<td>454</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>187</td>
<td>100</td>
<td>91.3</td>
<td>485</td>
<td>0%</td>
</tr>
<tr>
<td>Vowel</td>
<td>2</td>
<td>55</td>
<td>72.2</td>
<td>50.9</td>
<td>298</td>
<td>+0.3%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>99</td>
<td>80.7</td>
<td>56.5</td>
<td>152</td>
<td>+1.9%</td>
</tr>
<tr>
<td>10-parity</td>
<td>5</td>
<td>61</td>
<td>97.2</td>
<td>—</td>
<td>148</td>
<td>+1.7%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>73</td>
<td>97.6</td>
<td>—</td>
<td>108</td>
<td>+2.2%</td>
</tr>
<tr>
<td>NetTalk</td>
<td>15</td>
<td>3,476</td>
<td>86.3</td>
<td>70.5</td>
<td>13</td>
<td>+1.1%</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>6,926</td>
<td>92.9</td>
<td>73.1</td>
<td>9</td>
<td>+0.3%</td>
</tr>
</tbody>
</table>
Conclusions

- Escaping from local minima – Trace
  - Generate information bearing trajectory
  - Identify good starting points for local search

- Constraint satisfaction
  - Lagrangian formulation
  - Discrete Lagrangian formulation
Publications

- Constrained Problems

- Unconstrained Problems

- Non-Linear Discrete Optimization