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

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)

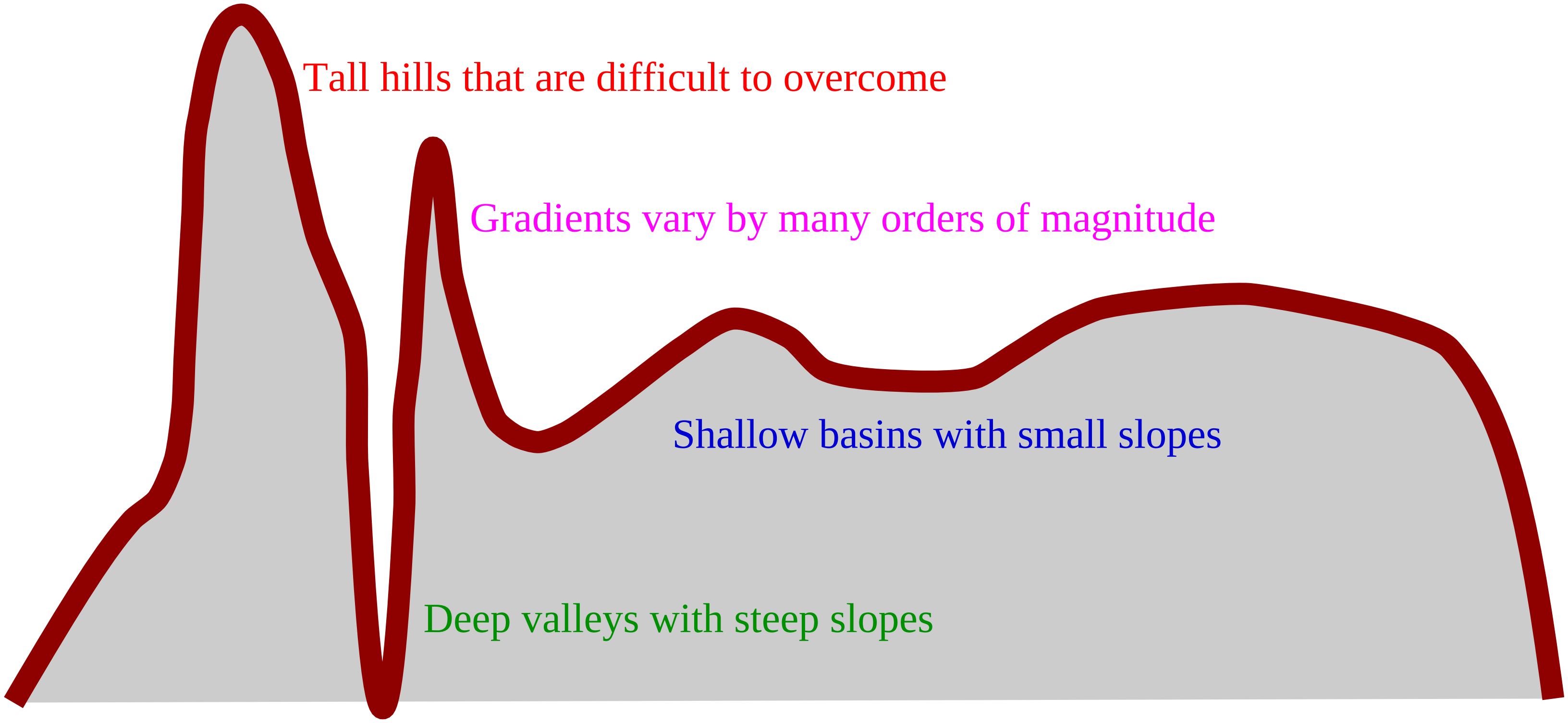
- Objective
  - Well defined/undefined
  - Linear/nonlinear/symbolic
- Objective measures
  - Deterministic/probabilistic
  - Resource measures
- Constraints
  - Hard/soft constraints
  - Linear/nonlinear/symbolic
  - Resource constraints

Characteristics of Search Algorithms

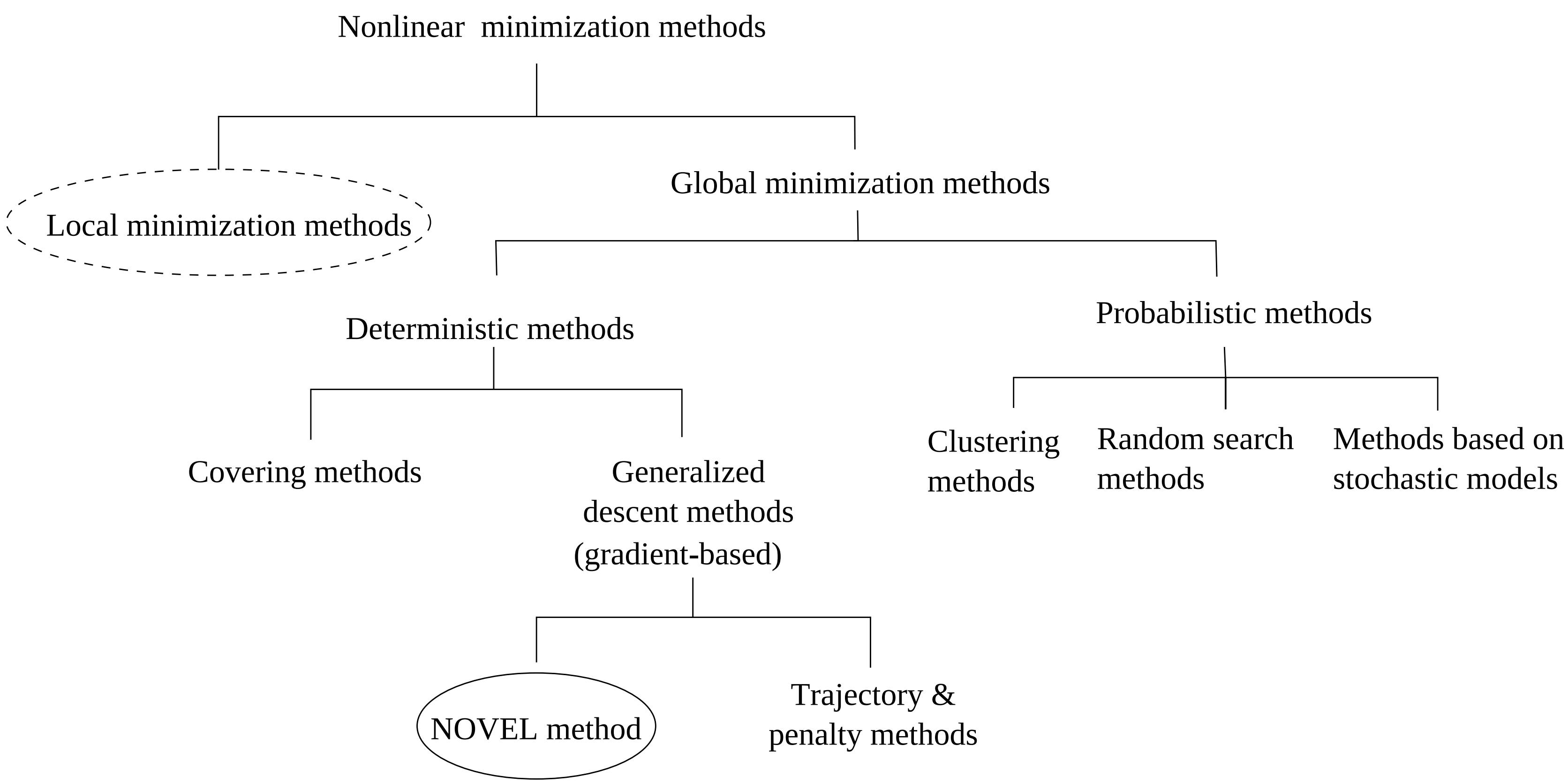
- Representation of search space
  - Search complexity
- Decomposition strategies
- Heuristic predictor or direction finder
  - Relaxation algorithms
- Mechanisms to help escape from local minima
- Mechanisms to handle constraints
- Stopping conditions
- Resource scheduling strategies

METHODS TO HELP ESCAPE FROM LOCAL MINIMA

Local Minima



Existing Methods to Help Escape from Local Minima



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

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0 \end{aligned}$$

- Lagrangian/Augmented Lagrangian functions

$$\begin{aligned} L(x, \lambda) &= f(x) + \lambda^T h(x) \\ \mathcal{L}(x, \lambda) &= f(x) + ||h(x)||_2^2 + \lambda^T h(x) \end{aligned}$$

- Sufficient conditions for optimality: System of differential equations

$$\begin{aligned} \bigtriangledown_x L(x, \lambda) = 0 \\ \bigtriangledown_\lambda L(x, \lambda) = 0 \end{aligned}$$

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 (λ carries no weight)
  - Gradient ascent in λ space ($\frac{d\lambda}{dt} = \nabla_\lambda \mathcal{L}(x, \lambda)$)
    - When constraints are violated, increase λ to increase weight of violation
- More effective than penalty methods in adjusting λ
- Handling inequality constraints
  - Slack variable method
  - MaxQ method

Discrete Lagrangian Methods

- Discrete optimization problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0, \quad x \in Z \end{aligned}$$

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

$$\begin{aligned} x_{k+1} &= x_k - \Delta_x L(x_k, \lambda_k) \\ \lambda_{k+1} &= \lambda_k + h(x_k) \end{aligned}$$

- 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

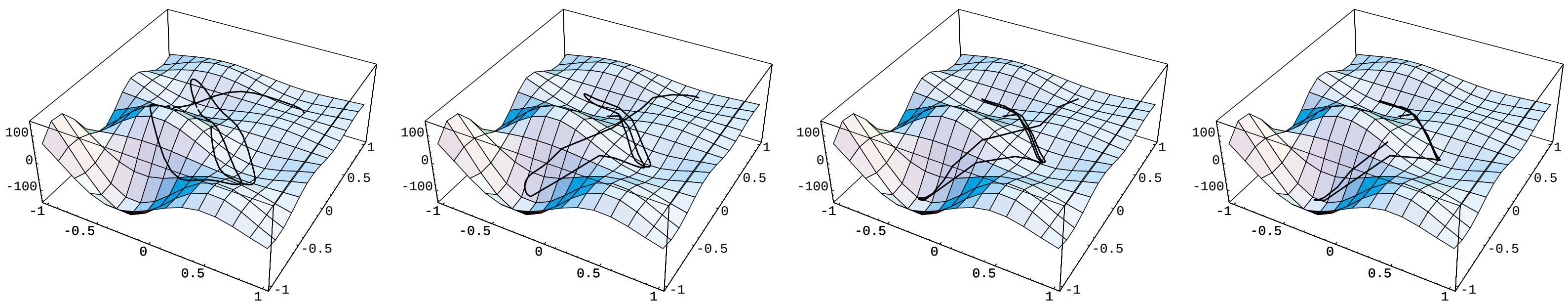
Features of NOVEL

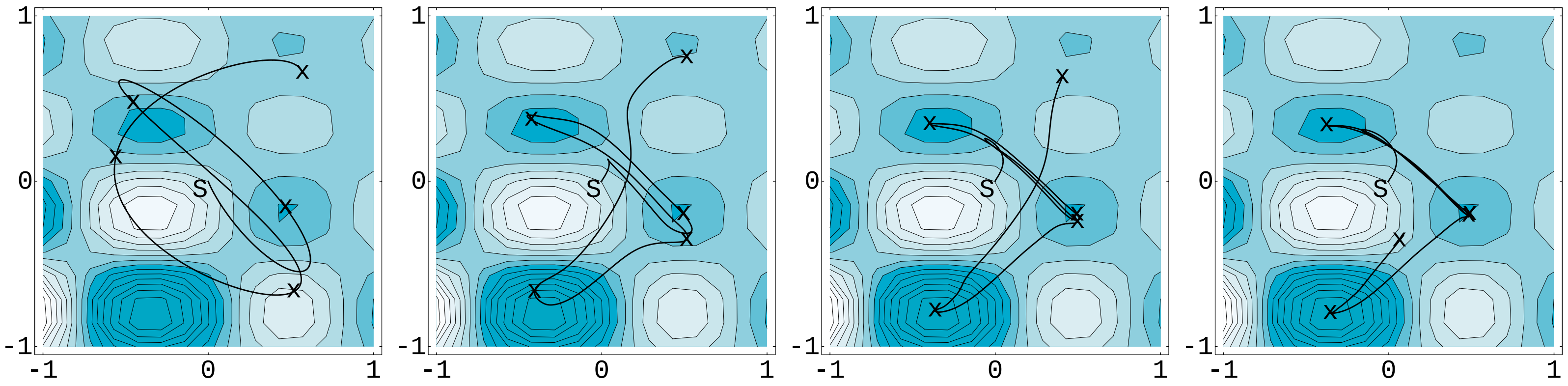
- Global search: locating promising regions
  - A user-defined trace function leading the search
  - Local minima attracting the search trajectory
- Local search
  - Gradient descent
  - Lagrangian search

A Simple Example

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

$$f_{l_3}(x) = \sum_{i=1}^{5} i\, cos[(i-1)x_1 + i] \sum_{j=1}^{5} j\, cos[(j+1)x_2 + j]$$




1
0
-1
-1
0
1
S
1
0
-1
-1
0
1
S
1
0
-1
-1
0
1
S
1
0
-1
-1
0
1
S

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

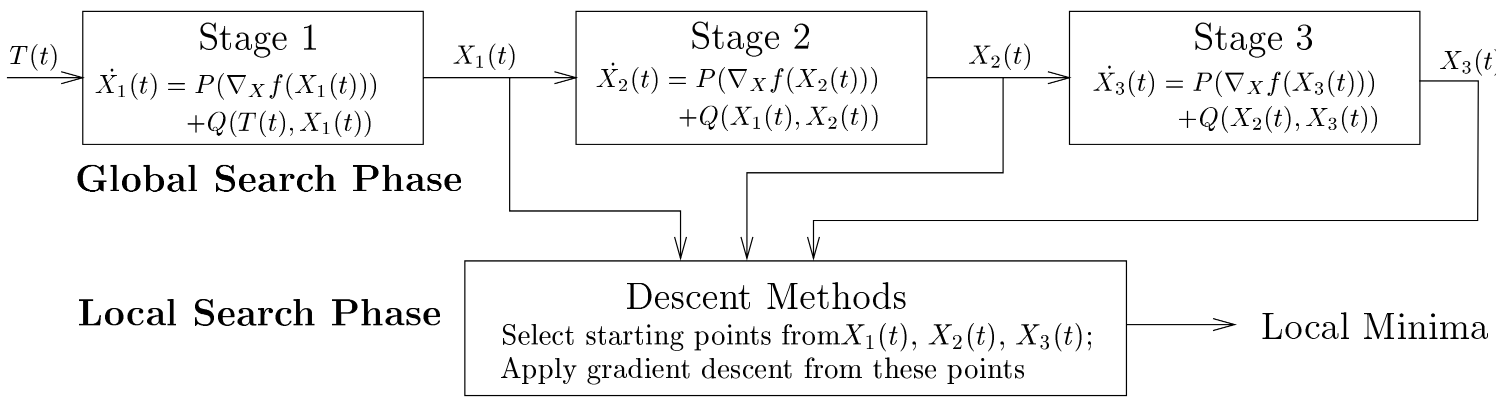
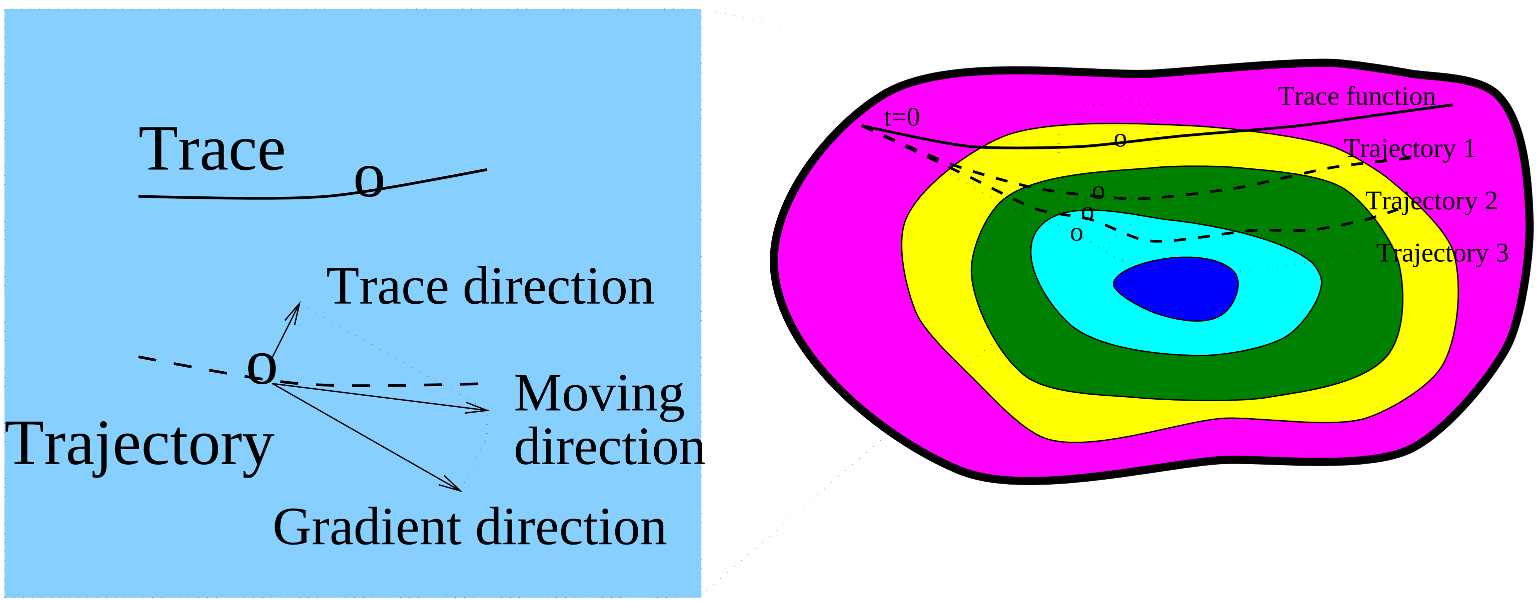


Illustration of Global Search Phase



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

$$= \rho\, sin \left[2\pi \left(\frac{t}{2}\right)^{0.95 - \frac{0.45(i-1)}{n})} + \frac{2\pi(i-1)}{n} \right]$$

  - t: autonomous variable
  - n: number of dimensions
  - i: i'th dimension
  - ρ: search range

Mathematical Formulation of Global Search Phase

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

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

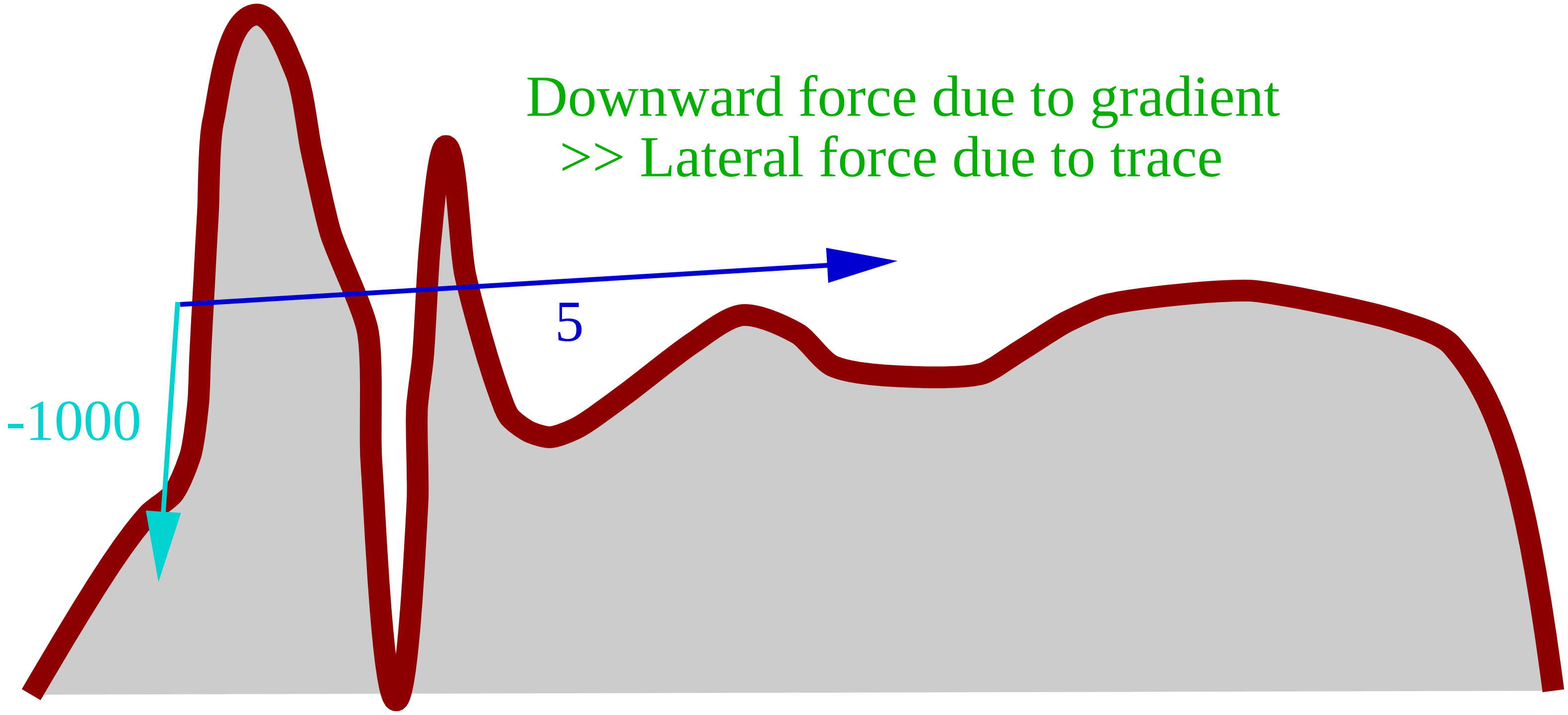
  - $f(X)$: Error function to be minimized
  - $\nabla_X f(X)$: Gradient of $f(X)$
  - $P(\nabla_X f(X(t)))$ enables gradient to attract the trajectory
  - $Q(T(t), X(t))$ allows trace function $T(t)$ to lead the trajectory

- One simple trajectory through variable space X

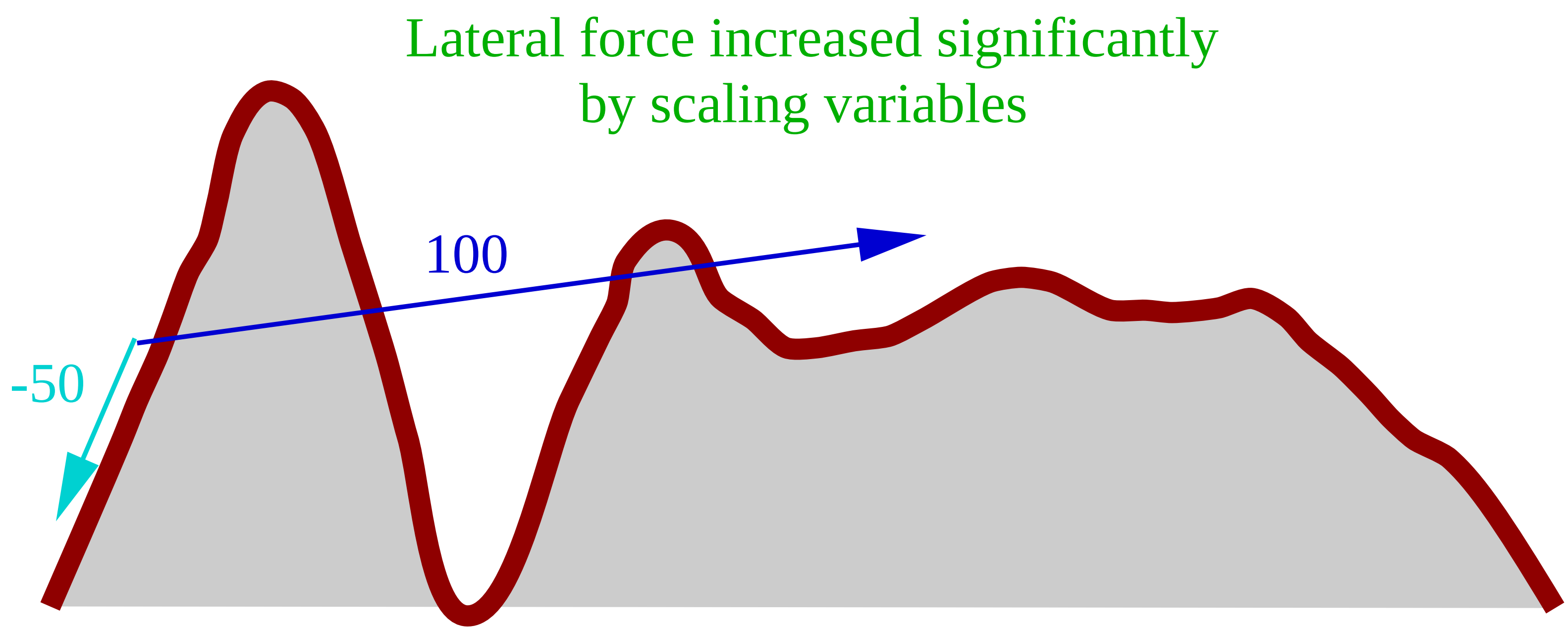
$$\frac{d\,X(t)}{dt} = -\mu_g \nabla_X f(X(t)) - \mu_t (X(t) - T(t))$$

  - μ_g and μ_t are positive real numbers

Insufficient Lateral Force



After 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

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

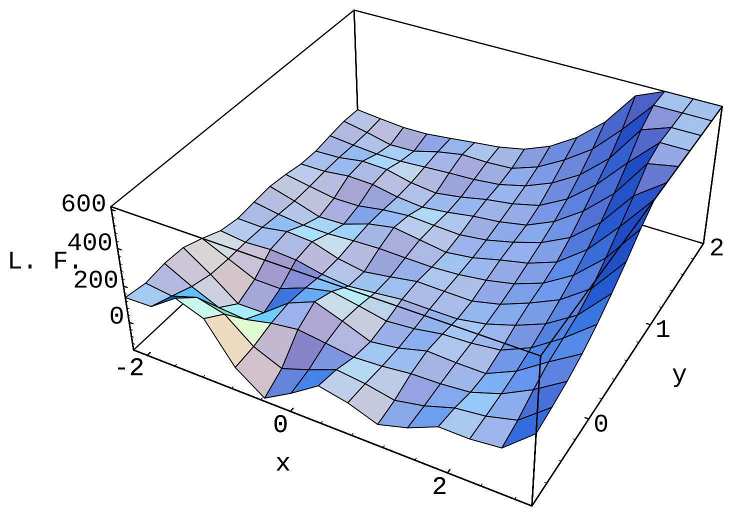
- Lyupunov function as stopping condition

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

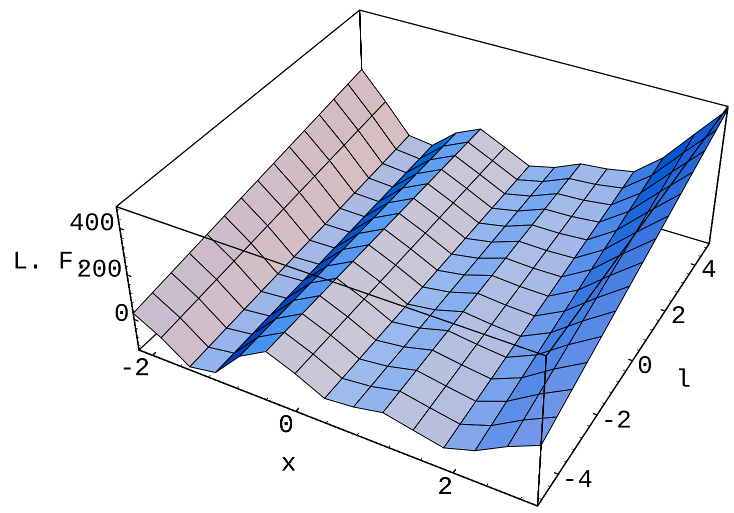
- Handling inequality constraint $g(x) \leq 0$

$$\left[max^2(0, \mu_i + g_i(X)) - \mu_i^2\right] = 0$$
$$[\mu_i max^{q_i}(0, g_i(X))] = 0$$

Example: Levy's No. 3 Function & Elliptic Constraint

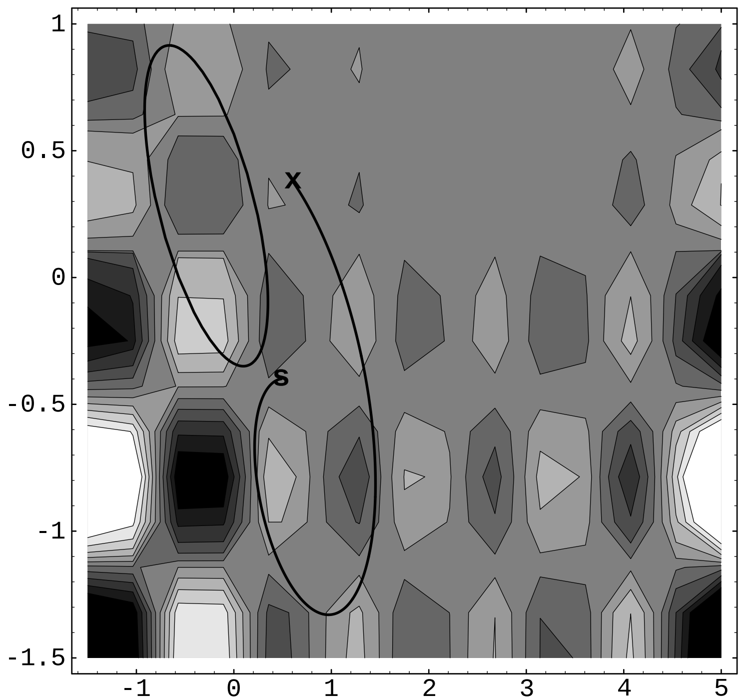


x-y plot

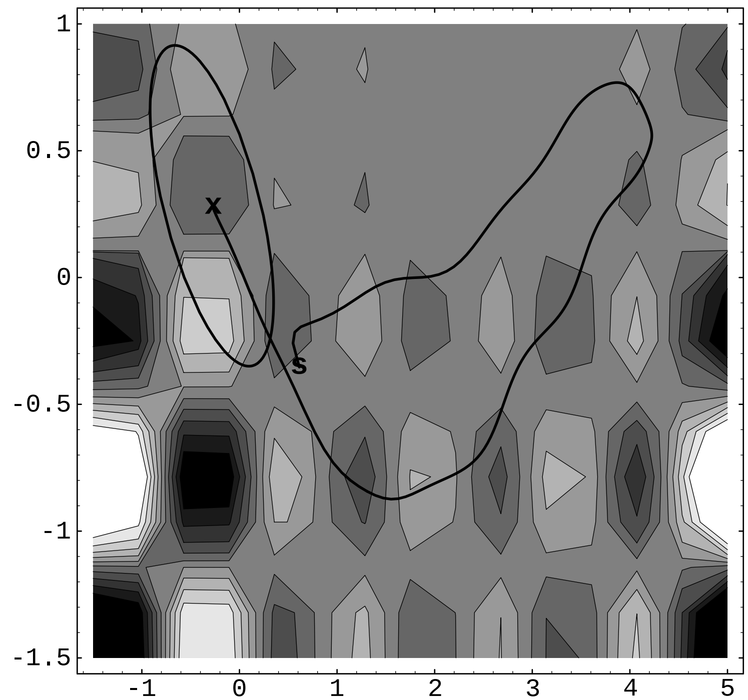


x-λ plot

Example (cont'd)



Objective, constraint, & trace



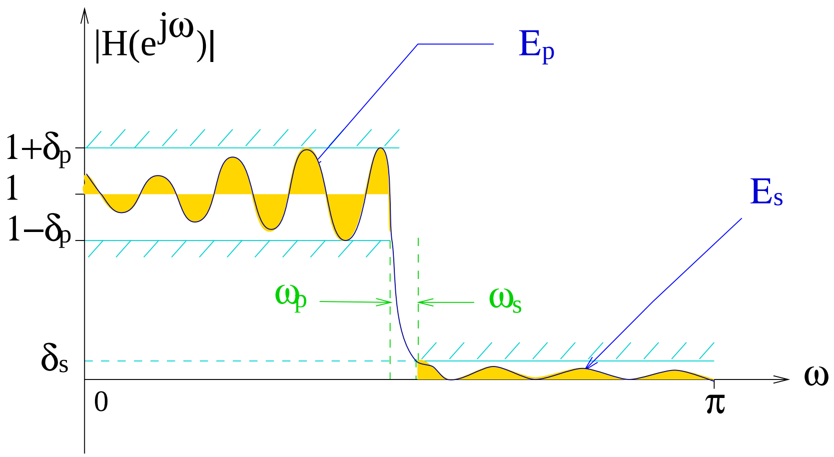
Objective, constraint, & trajectory

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

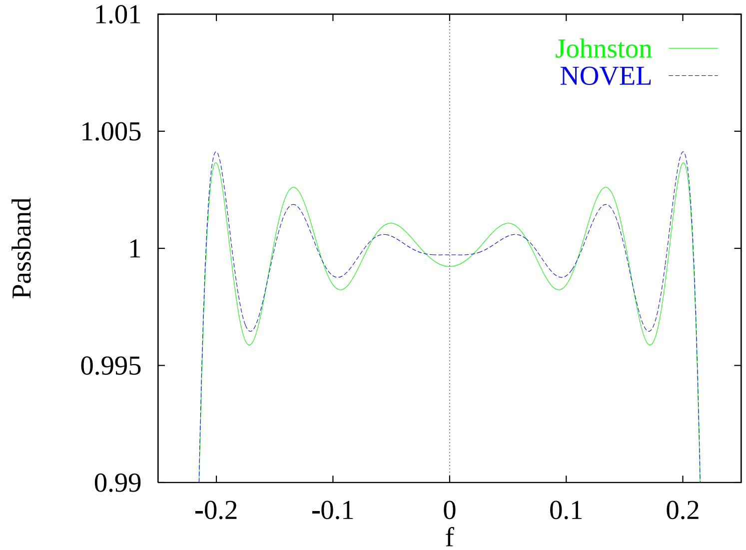
APPLICATION 2: QMF FILTER-BANK DESIGN

QMF Filter Bank Design

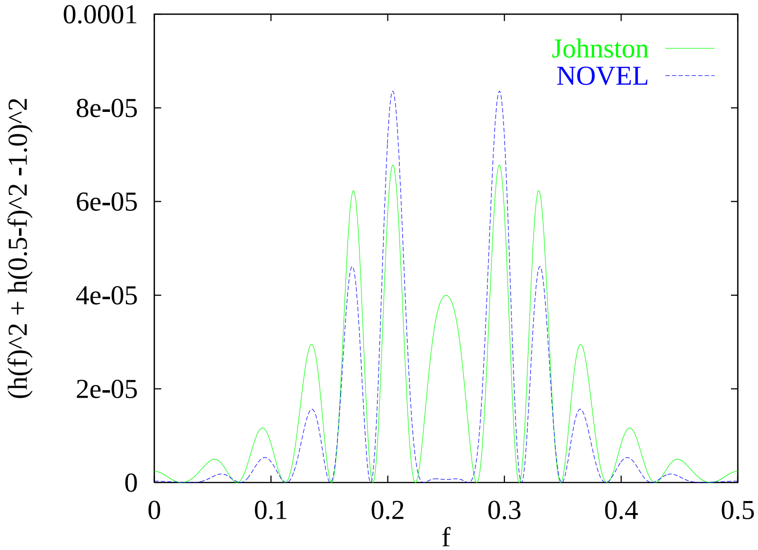
Filter	Design Objectives
Overall Filter Bank	Min amplitude distortion
	Min aliasing distortion
	Min phase distortion
Single Filter	Min stopband ripple (δ_s)
	Min passpand ripple (δ_p)
	Min transition band error (E_t)
	Min stopband energy (E_s)
	Max passband flatness (E_p)



Example: 24D QMF Design



Passband frequency response

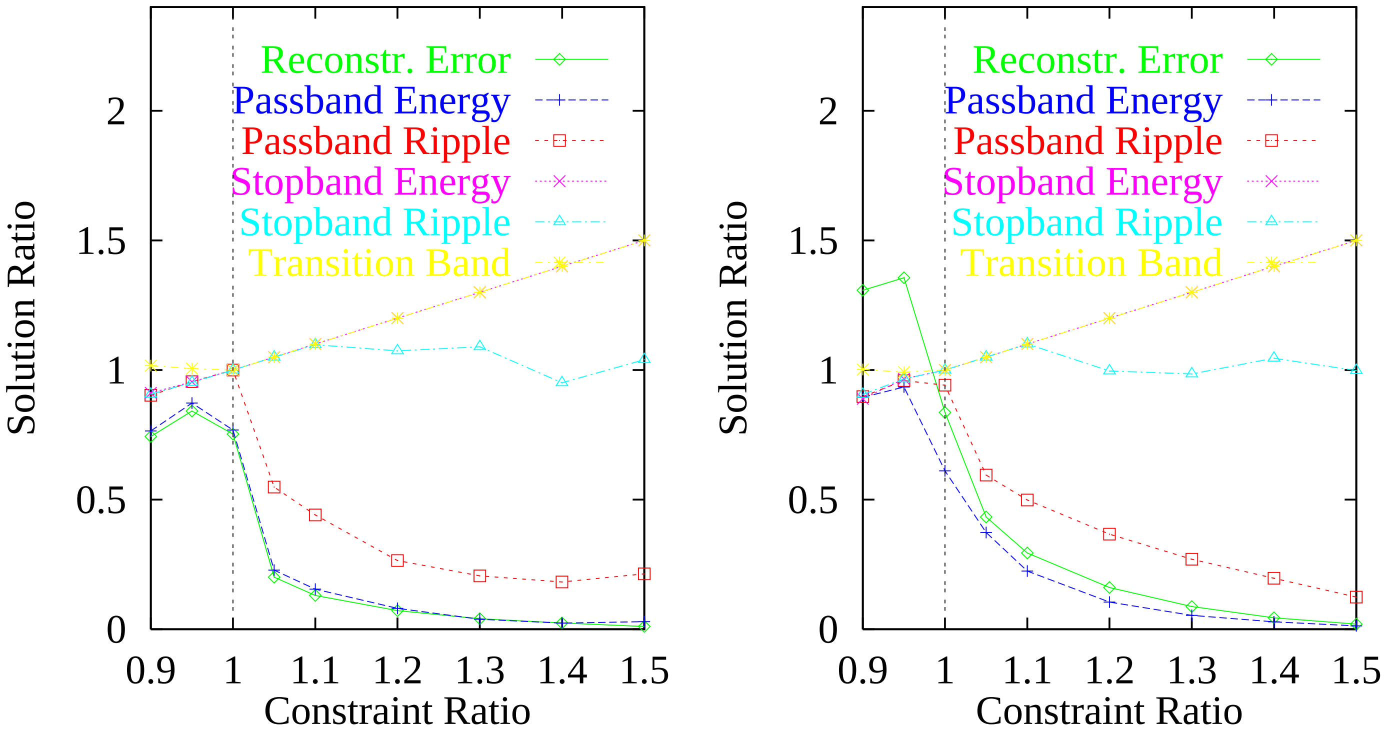


Reconstruction error

Initial Results: Better Designs

Filter-type	E_r	δ_p	E_p	δ_s	E_s	T_r
16a	0.986	1.000	0.858	1.000	1.000	1.000
16b	0.985	1.000	0.894	1.000	1.000	1.000
16c	0.820	1.000	0.926	1.000	1.000	1.000
24b	0.964	1.000	0.802	1.000	1.000	1.000
24c	0.893	0.977	0.588	1.000	1.000	1.000
24d	0.753	1.000	0.768	1.000	1.000	1.000
32c	0.959	1.000	0.748	1.000	1.000	1.000
32d	0.870	1.000	0.802	1.000	1.000	1.000
32e	0.712	1.000	0.896	1.000	1.000	1.000
48c	0.787	0.972	0.793	0.970	0.999	1.000
48d	0.947	1.000	0.756	0.999	1.000	1.000
48e	0.852	1.000	0.842	1.000	1.000	1.000
64d	0.821	0.981	0.767	0.963	0.999	1.000
64e	0.843	1.000	0.749	1.000	1.000	1.000

Initial Results: Different Constraints



24D

32D

APPLICATION 3: DISCRETE SATISFIABILITY

Satisfiability (SAT) Problem

- Given
  - a set of m clauses $C_1, C_2, \cdots, C_m$ on n variables

$$X = (x_1, x_2, \cdots, x_n) \qquad x_i \in \{0, 1\}$$

  - 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

$$\text{Find } x \text{ such that} \quad U_i(X) = 0 \quad i = 1, \ldots, m$$

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

  – 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

$$
\begin{aligned}
c_i(x) &= \prod_{j=1}^{m} a_{i,j}(x_j) \\
a_{i,j}(x_j) &= \begin{cases} (1-x_j)^2 & \text{if } x_j \text{ in } C_i \\ x_j^2 & \text{if } \bar{x}_j \text{ in } C_i \\ 1 & \text{otherwise} \end{cases} \\
\text{Objective}: \ min & \sum_i c_i(x)
\end{aligned}
$$

  - Gu's UniSAT model
  - Local search methods: gradient descent, conjugate gradient, Quasi-Newton
  - Computationally expensive

Alternative Formulations (cont'd)

- Continuous constrained formulation

$$\begin{array}{ll} \displaystyle\min_{x \in E^m} & f(x) = \displaystyle\sum_{i=1}^{n} c_i(x) \\ \text{subject to} & c_i(x) = 0 \quad \forall i \in \{1, 2, \ldots, n\} \end{array}$$

  - Smooth out local minima in discrete space
  - Lagrangian methods
  - Very expensive

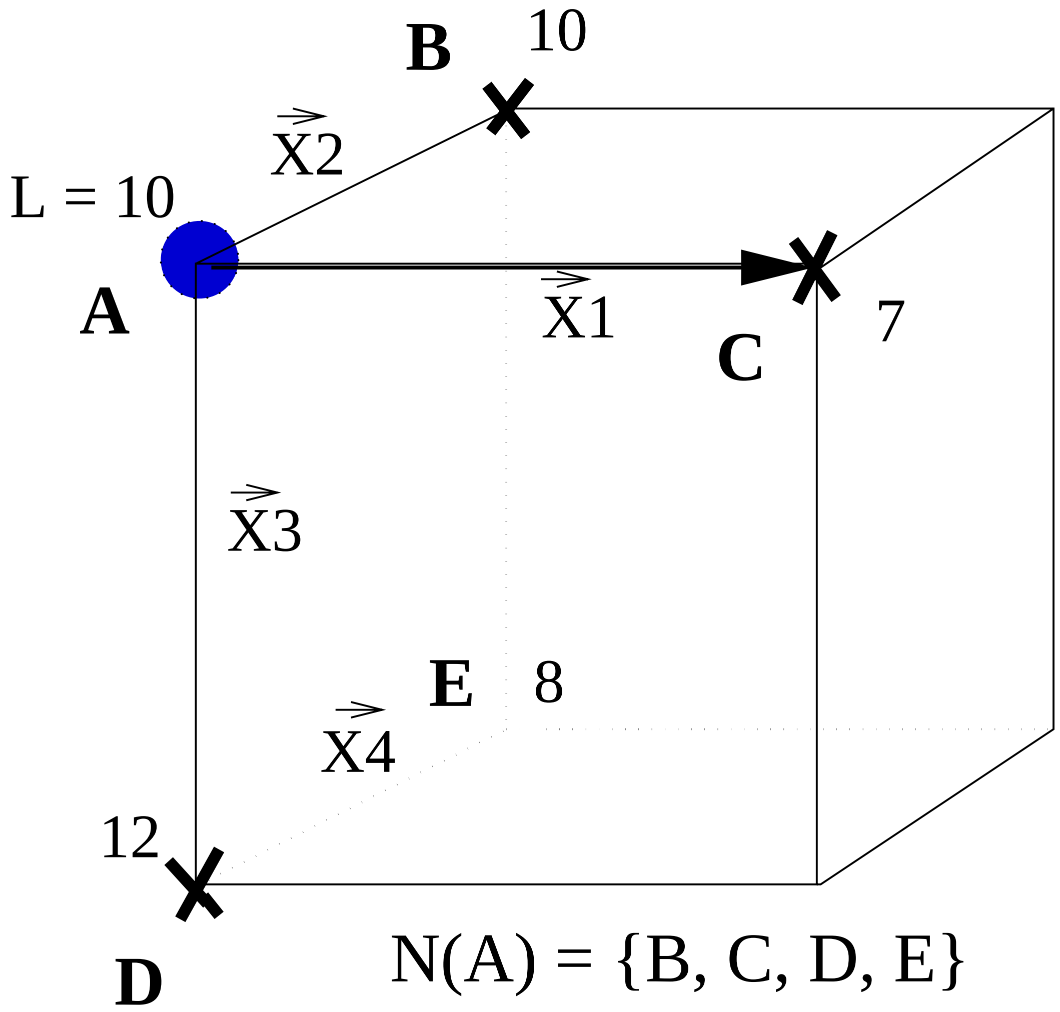
Our SAT Formulation

- Constrained optimization with artificial (discrete) objective

$$\begin{aligned} \min \quad & N(X) = \sum_{i=1}^{m} U_i(X) \\ \text{subject to} \quad & U_i(X) = 0 \quad i = 1, \ldots, m \end{aligned}$$

  - 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 $\Delta_X L(X, \lambda)$



An Implementation to solve SAT Problems

Set initial point x
while x is not a solution, *i.e.*, $N(x) > 0$
    while $\Delta_x L(x, \lambda) \neq 0$
        update x: $x \longleftarrow x - \Delta_x L(x, \lambda)$
    end while
    update λ: $\lambda \longleftarrow \lambda + c \times U(x)$
end while

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

Problem Id	No. of Var.	No. of Clauses	DLM Version 2			WSAT	GSAT	DP
			SS 10/51	Challenge	# Iter.			
ssa7552-038	1501	3575	0.228	0.3	7970	2.3	129	7
ssa7552-158	1363	3034	0.088	0.1	2169	2	90	*
ssa7552-159	1363	3032	0.085	0.1	2154	0.8	14	*
ssa7552-160	1391	3126	0.097	0.1	3116	1.5	18	*

Problem Id.	No. of Var.	No. of Clauses	DLM Version 2			GSAT	Integer Prog.	SA
			SS 10/51	Challenge	# Iter.			
ii16a1	1650	19368	0.122	0.128	819	2	2039	12
ii16b1	1728	24792	0.265	0.310	1546	12	78	11
ii16c1	1580	16467	0.163	0.173	797	1	758	5
ii16d1	1230	15901	0.188	0.233	908	3	1547	4
ii16e1	1245	14766	0.297	0.302	861	1	2156	3

Problem Identification	No. of Var.	No. of Clauses	DLM Version 3		GSAT	
			Time	Success	Time	Success
g125.17	2125	66272	1390.32	10/10	264.07	7/10
g125.18	2250	70163	3.197	10/10	1.9	10/10
g250.15	3750	233965	2.798	10/10	4.41	10/10
g250.29	7250	454622	1219.56	9/10	1219.88	9/10

Results On Difficult But Satisfiable DIMACS Benchmarks

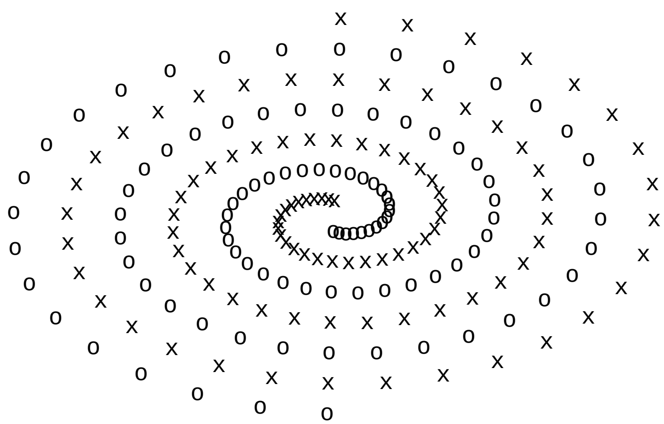
Prob. Id.	Succ. Ratio	Time in CPU seconds			Prob. Id.	Succ. Ratio	Time in CPU seconds		
		Avg.	Min.	Max.			Avg.	Min.	Max.
par8-1	10/10	4.780	0.133	14.383	par16-1-c	10/10	398.1	11.7	1011.9
par8-2	10/10	5.058	0.100	13.067	par16-2-c	10/10	1324.3	191.0	4232.3
par8-3	10/10	9.903	0.350	21.150	par16-3-c	10/10	987.2	139.8	3705.2
par8-4	10/10	5.842	0.850	16.433	par16-4-c	10/10	316.7	5.7	692.66
par8-5	10/10	14.628	1.167	34.900	par16-5-c	10/10	1584.2	414.5	3313.2
hanoi4	1/10	682.6	682.6	682.6	f1000	10/10	126.8	4.4	280.7
f600	10/10	16.9	2.1	37.2	f2000	10/10	1808.6	174.3	8244.7

- Still cannot solve 16 satisfiable DIMACS benchmark problems
  - *par16-1* thru *par16-5*
  - *par32-1* thru *par32-5*
  - *par32-1-c* thru *par32-5-c*
  - *hanoi5*

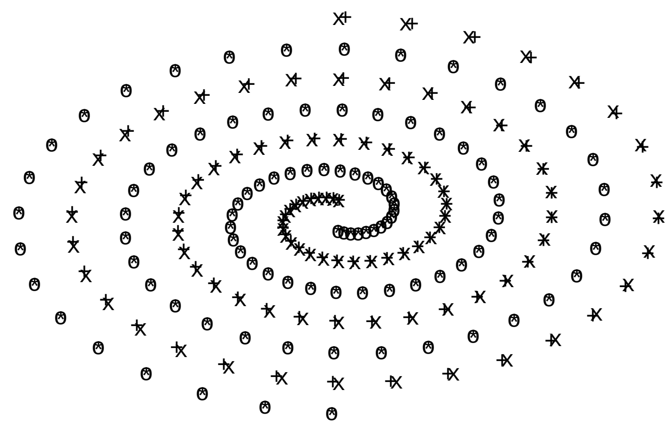
APPLICATION 4: FEEDFORWARD NEURAL-NETWORK LEARNING

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



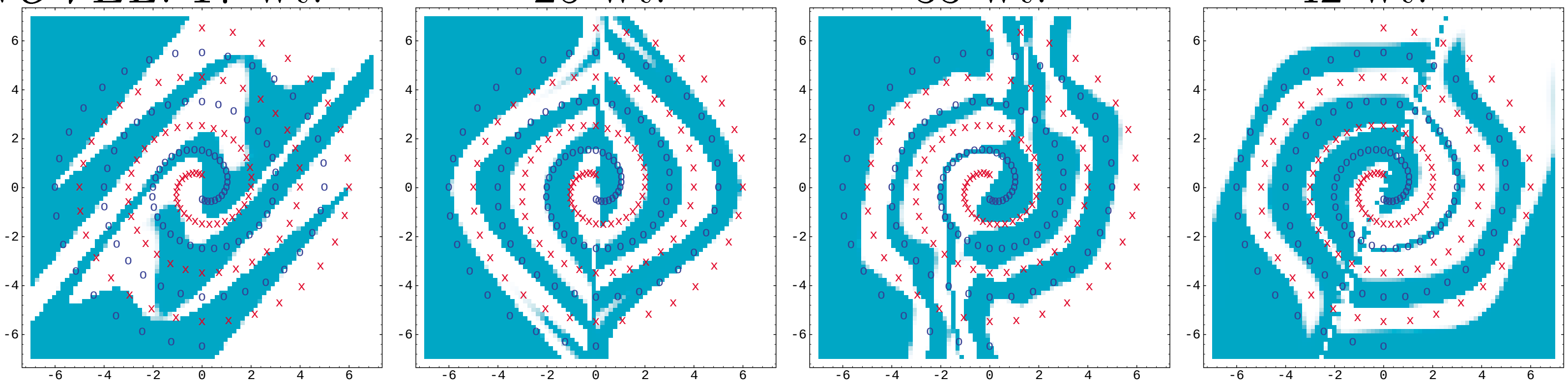
194 training patterns



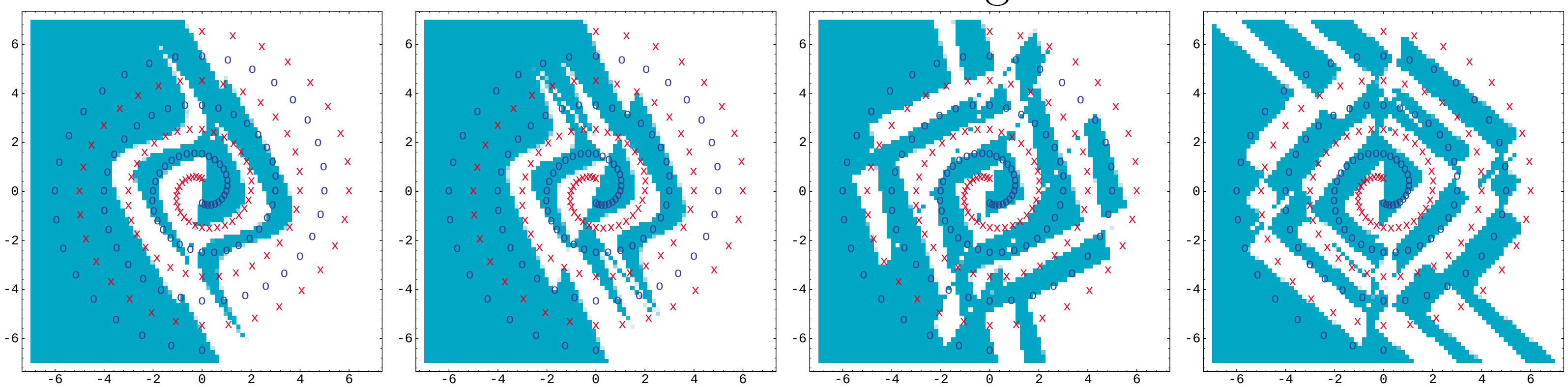
194 testing patterns

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

Problems	# of H.U.	# of Wts.	*TN-MS*			*NOVEL*			*TN-MS + NOVEL*			CPU time limits
			Correct %		# of restarts	Correct %		# time units	Correct %		# time units	
			training	test		training	test		training	test		
Sonar	2	125	98.1	90.4	454	0%	**+3.8%**	191	0%	**+1.9%**	226	1000 sec
	3	187	100	91.3	485	0%	**+1%**	291	0%	**+1%**	315	2000 sec
Vowel	2	55	72.2	50.9	298	**+0.3%**	-1.8%	131	**+1.3%**	-0.3%	203	2 hours
	4	99	80.7	56.5	152	**+1.9%**	**+1.3%**	41	**+0.5%**	**+0.6%**	168	2 hours
10-parity	5	61	97.2	—	148	**+1.7%**	—	51	0%	—	49	2000 sec
	6	73	97.6	—	108	**+2.2%**	—	62	0%	—	44	3000 sec
NetTalk			*Pattern-wise BP*			*NOVEL*			*Pattern-wise BP + NOVEL*			
	15	3,476	86.3	70.5	13	**+1.1%**	**+2.2%**	11	**+2.7%**	-0.1%	11	3 hours
	30	6,926	92.9	73.1	9	**+0.3%**	-0.6%	4	**+1.8%**	-0.8%	7	4 hours

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
  - "Trace-Based Methods for Solving Nonlinear Global Optimization and Satisfiability Problems," B. W. Wah and Y. J. Chang, *J. of Global Optimization*, (to appear) 1996.
  - "Handling Inequality Constraints in Continuous Nonlinear Global Optimization," T. Wang and B. W. Wah, *Proc. Society for Design and Process Science Conference*, December 1996.
  - "Global Optimization Design of QMF Filter Banks," B. W. Wah, Y. Shang, T. Wang and T. Yu, *Proc. IEEE Midwest Symposium on Circuits and Systems*, August 1996.
- Unconstrained Problems
  - "Global Optimization for Neural Network Training," Y. Shang and B. W. Wah, *IEEE Computer*, vol. 29, No. 3, March 1996, pp. 45-54.
  - Neural-Network Training Software: Sun Sparc object code, Y. Shang and B. W. Wah, Released: May 27, 1996.
  - "A Global Optimization Method for Neural Network Training," Y. Shang and B. W. Wah, Proc. 1996 IEEE Int'l Conf. on Neural Networks (Plenary, Panel and Special Sessions), pp. 7-11, June 1996.
- Non-Linear Discrete Optimization
  - "A Discrete Lagrangian-Based Global-Search Method for Solving Satisfiability Problems," B. W. Wah and Y. Shang *Proc. DIMACS Workshop on Satisfiability Problem: Theory and Applications*, Ed: Ding-Zhu Du, Jun Gu, and Panos Pardalos, American Mathematical Society, March 1996.