OQNLP: a Scatter Search Multistart Approach for Solving Constrained Non-Linear Global Optimization Problems

Zsolt Ugray, The University of California at Riverside, MIS Dept.
Leon Lasdon, The University of Texas at Austin , MSIS Dept.
John Plummer, The University of Texas at Austin , MSIS Dept.
Jim Kelly, OptTek Systems Inc, www.opttek.com

OQNLP-A Multi-start Search Method

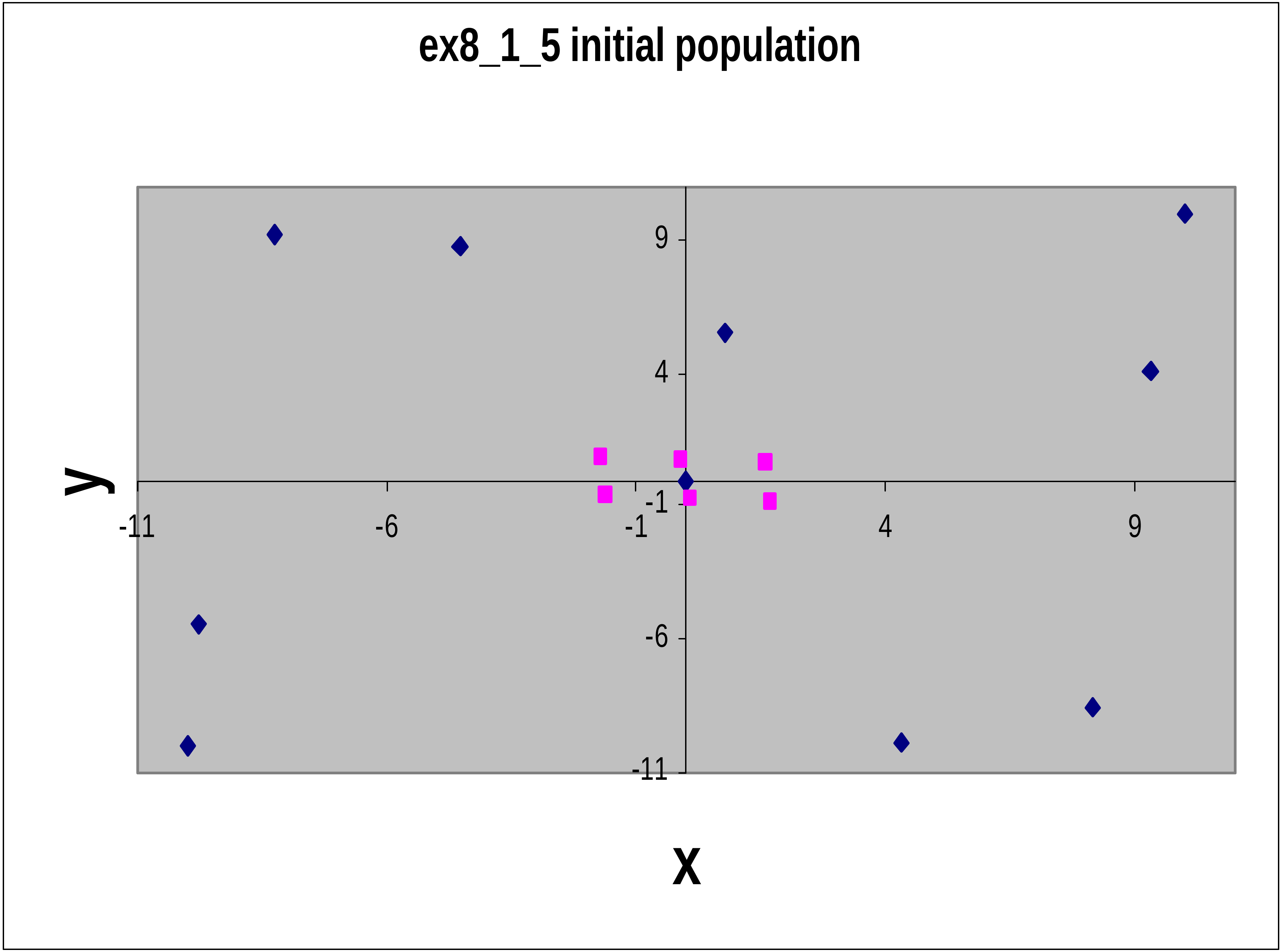
- Intended for nonlinearly constrained, smooth, non-convex NLP's and MINLP's
- Combines the OptQuest Callable Library (a scatter search code) of Glover, Laguna, Kelly with a local NLP solver
- Currently uses LSGRG2 , a sparse gradient-based GRG code, as the local NLP solver in callable mode
- GAMS version can call any GAMS NLP solver (conopt, minos, snopt, lsgrg2)
- Written in C

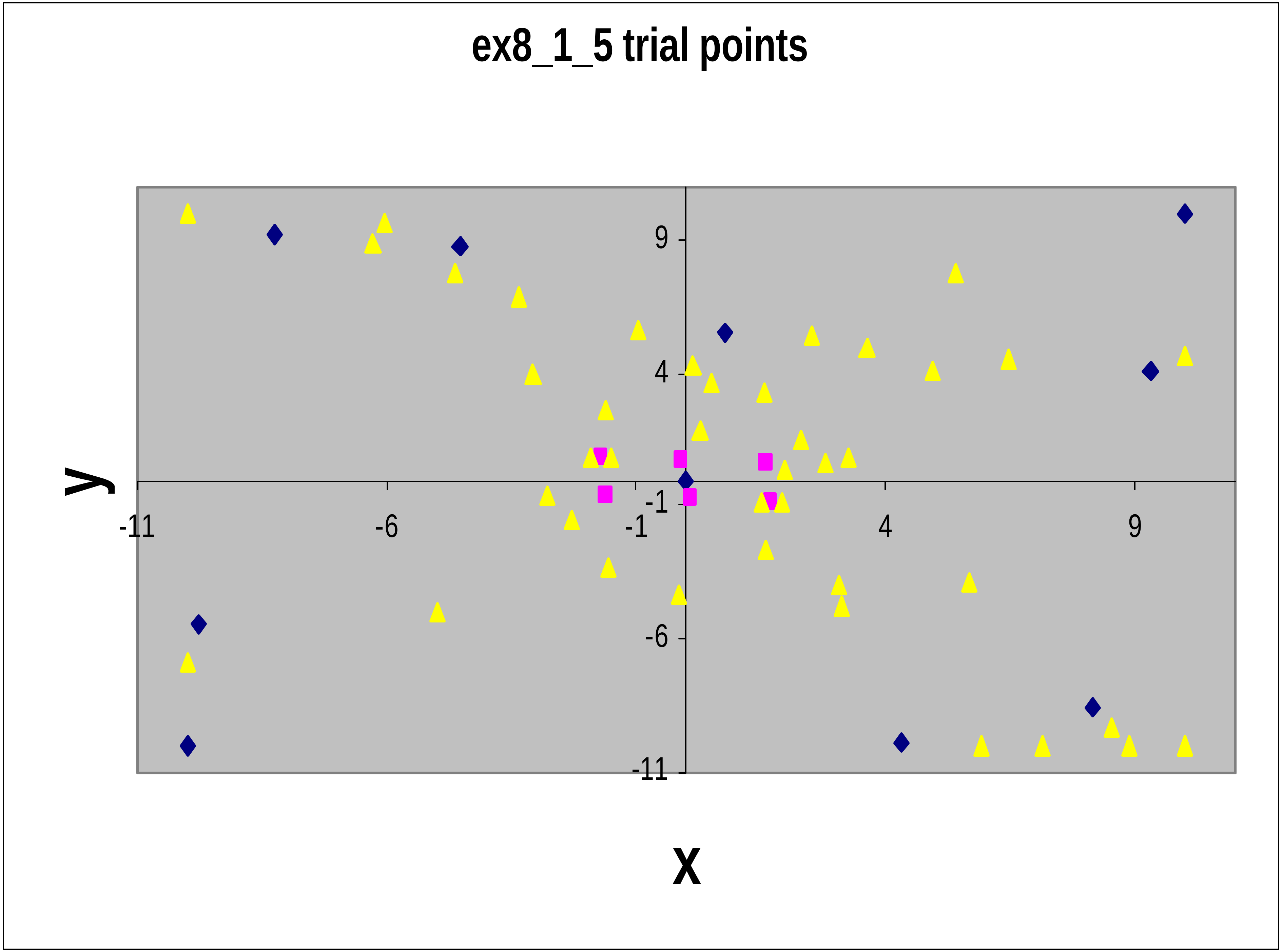
Other multistart methods:multi- level single linkage

- Developed by Rinnooy Kan and Timmer for smooth unconstrained problems, 1987-89.
- (1) Generate N uniformly distributed points, retain p*N best, where $0<p<1$.
- (2) Start local solver from each retained point, unless there is a better point within a distance r (decreases).
- (3) add N more uniformly distributed points, go to (1)
- They Prove: total number of local searches finite and each local optimum located with probability one.
- Recent implementation by Fylstra of Frontline Systems for constrained problems, using L1 exact penalty function.

OptQuest (OQ) and OQ Callable Library

- OQ is the Glover-Laguna-Kelly implementation of scatter search
- Option within the **Crystal Ball** Excel add-in for Monte-Carlo simulation, and within **ARENA** for discrete event simulation
- Handles MINLP’s
- OQ callable library:
  - set up problem
  - define variables, bounds, constraints, requirements
  - create initial population
  - for N iterations:
    - get trial solution
    - evaluate objective
    - put trial solution, objective back to database
    - regenerate population if necessary


ex8_1_5 initial population
9
4
-1
-6
-11
-11
-6
-1
4
9
y
x



ex8_1_5 trial points
y
x
9
4
-1
-6
-11
-11
-6
-1
4
9

Search Method Drawbacks

- If feasible region is narrow (e.g. equality constraints) then very difficult for OQ (and other search methods) to find a feasible solution
- Often finds good solution fast, but requires many additional function evaluations to get high accuracy
- Gradient-based NLP solvers (GRG,SQP,SLP) are much better at getting feasible and attaining high accuracy, but find “nearest” local optimum.

When to Start the NLP Solver (*L*)

- An *L* call is expensive-many function evaluations
- Ideally: start once in the basin of attraction of each local solution
- Don't start from a trial point if
  - too close to a previously found local solution(*distance filter)*
  - exact penalty function value is too large (*merit filter*)

Distance Filter

- Store max distance traveled to each local optimum

- If the distance from a candidate starting point to any local opt is < *distfactor**maxdist, don't start *L* (default *distfactor = 0.75)*

- This assumes regions of attraction are spherical, and maxdist is a good estimate of the radius

Merit Filter

- Use L1 exact penalty , P(x,w), as merit function
  - P(x,w)=obj(x)+sum(i,wi*infeasi(x))
  - wi > max abs Lagrange multiplier for constraint i over all local solutions
- Don't start *L* from candidate points which have value for P > threshold
- Initially threshold=best P value over all candidate points so far
- If P > threshold more than 20 consecutive times,

  threshold = threshold +*threshfactor**(1+abs(threshold))

  (default *threshfactor = 0.2)*

Dynamic Filters

- **Distance Filter**
  - actual basins of attraction partition the hyper-rectangle
  - adjust radii of spherical basin models so that
    - r(i)+r(j)<=d(i,j)
- **Merit Filter**
  - at each rejected point x(k), compute factor,f(k), such that

    threshold +*f(k)**(1+abs(threshold))=penval(k)
  - if merit filter rejects *waitcycle* consecutive points, set factor for increasing threshold to

    factor = max[user factor, min(f(k))]
- **Effects still being evaluated**
  - somewhat more solver calls
  - harder to measure possible increase in robustness or reduction in solver calls to find global solution.

OQNLP Algorithm-Stage 1

INITIALIZATION

Read_Problem_Parameters (size, bounds, starting point);

Setup_OptQuest_Parameters (size, iteration limits, population, accuracy, etc);

Initialize_OptQuest_Population;

Call *L*(user starting point, local solution)

STAGE 1: INITIAL OPTQUEST ITERATIONS AND FIRST CALL to *L*

WHILE (stage 1 iterations remain) **DO** {

Get (trial solution from OptQuest);

Evaluate (objective and nonlinear constraint values at trial solution,);

Put (trial solution , objective and constraint values to OptQuest database); }

ENDDO

Get_Best_Point_from_OptQuest_database (starting point);

Call_*L* (starting point, local solution);

threshold = Penalty value of local solution;

OQNLP-Stage 2

STAGE 2: MAIN ITERATIVE LOOP

WHILE (stopping criteria not met) **DO** {

Get (trial solution from OptQuest);

Evaluate (objective and nonlinear constraint values at trial solution,);

Put (trial solution, objective and constraint values to OptQuest database);

Calculate_ Penalty_ Function (trial solution, P_1);

IF (distance and merit filter criteria are satisfied) **THEN** {

Call_*L* (trial solution, local solution);

Analyze_Solution (*L* Termination Condition);

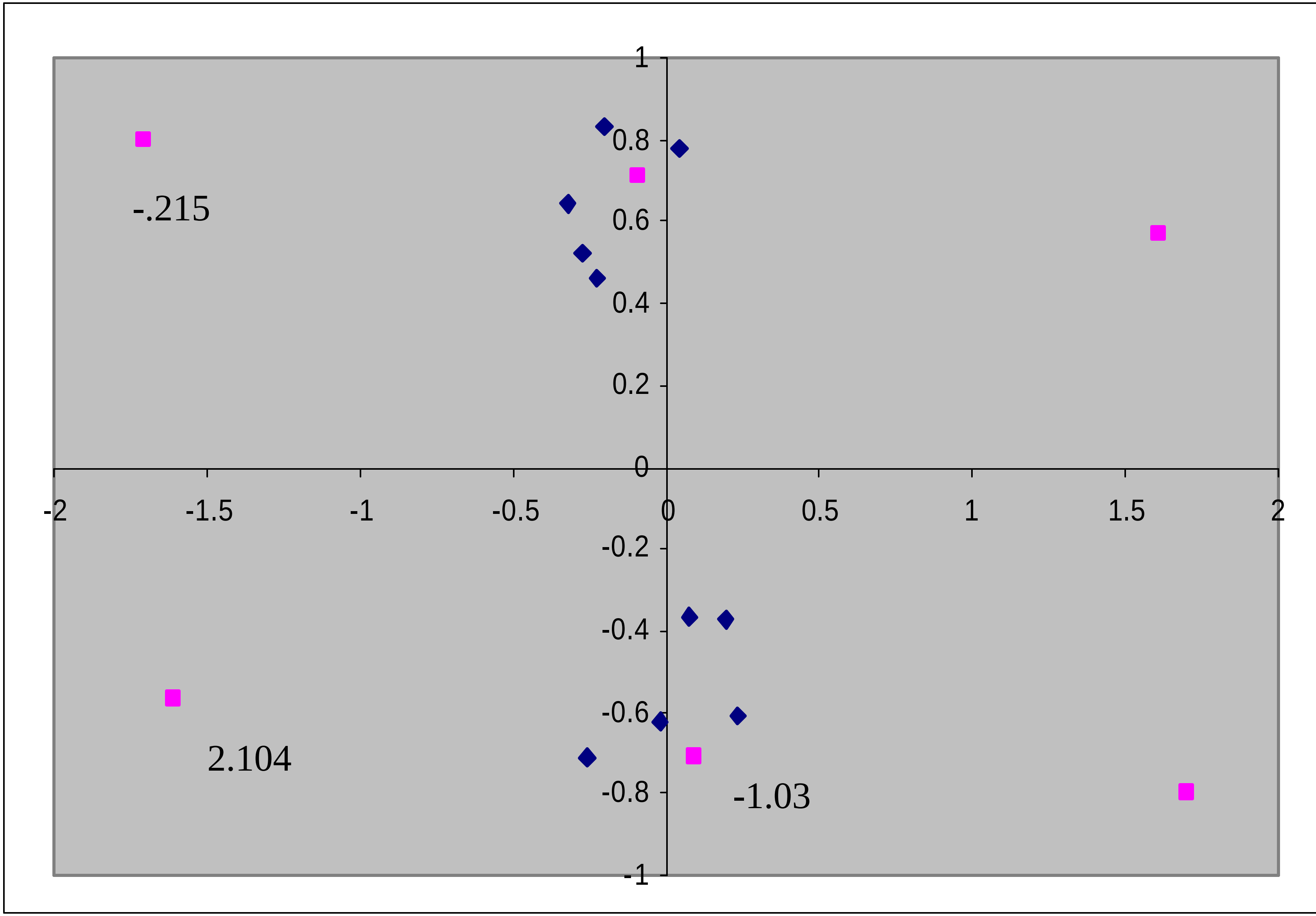
Update_Local_Solutions_Found;

Update_Largest_Lagrange_Multipliers_Found;

ELSE IF (Pen> threshold for *waitcycle* consecutive iterations) increase *threshold*

ENDDO

Points where GRG is called-200 initial OQ itns



Why no starting points near other locals

- Best point from 200 initial OQ itns has f = -.857, lower than other 2 locals, and in basin of attraction of one of the globals.
- This is point for initial *L* call, and sets initial threshold
- Even when threshold is increased, no other starting point is near the other locals, despite the effect of the distance filter.
- When no initial OQ itns, same thing happens.
- When threshold raised every 5 itns instead of 20, same thing happens, but *L* called 22 times rather than 10.

Handling Discrete Variables

- Mode 1:
  - *L* treats discrete variables as fixed, at values provided by OQ
  - OQ varies both discrete and continuous variables
  - *L* solutions not returned to OQ
  - Apply distance and merit filters only if discrete variables are the same
- Mode 2:
  - as above, but OQ varies only the discrete variables
  - Obj value returned to OQ is optimal value found by *L*
  - equivalent to OQ solving the discrete problem, where *L* optimizes over the continuous ones

GAMS Interface

- Motivation: Large Set of Test Problems Coded in GAMS by Floudas et.al., best known sol available
  - downloadable at http://titan.princeton.edu/TestProblems/
  - Most from Chemical process design or operation
  - 142+2N   problems
  - See book: Handbook of Test Problems in Local and Global Optimization, by Floudas et. al., Kluwer Academic Publishers, ISBN 0-7923-5801-5
- most small but some with over 100 variables, a few with over 1000.
- Most arise from chemical engineering
- Uses GAMS C Language Library Routines

Series	problems	max vars	max discrete vars	max linear constraints	max nonlinear constraints	Problem Type
EX2_1_x	14	24	0	10	0	concave QP (min)
EX3_1_x	4	8	0	4	6	quad obj and const
EX4_1_x	9	2	0	0	2	obj or cons polynomial
EX5_2_x	2	32	0	8	11	bilinear-pooling
EX5_3_x	2	62	0	19	34	distillation col sequencing
EX5_4_x	3	27	0	13	6	heat exch network
EX6_1_x	4	12	0	3	6	gibbs free energy min
EX6_2_x	10	9	0	3	0	gibbs free energy min
EX7_2_x	4	8	0	3	12	gen geometric prog
EX7_3_x	6	17	0	10	11	robust stability ana.
EX8_1_x	8	6	0	0	5	small unconstrained,nl constrained
EX8_2_x	5	55	0	6	75	batch plant design-uncertainty
EX8_3_x	14	141	0	43	65	CSTR network synthesis
EX8_4_x	8	62	0	0	40	constrained least squares
EX8_5_x	6	6	0	2	2	min tan plane distance
EX8_6_1	N from 4 to 147	3N	0	0	0	Lenard-Jones energy min
EX8_6_2	N from 5 to 80	3N	0	0	0	Morse energy min
EX9_1_x	10	29	0	27	5	bilevel LP
EX9_2_x	9	16	0	11	6	bilevel QP
EX12_2_x	6	11	8	9	4	MINLP
EX14_1_x	9	10	0	4	17	infinity norm solution of equations
EX14_2_x	9	7	0	1	10	infinity norm solution of equations
Total	142+2N					

Floudas Problem Summary

Parameters of base case

- Use 200 stage 1 iterations and 1000 total,
- OQ does not generate trial points which satisfy the linear constraints (greatly increases run times)
- Use LSGRG2 NLP solver for all but largest problems, SNOPT for these (Conopt and Lsgrg2 have infeasibility problems)
- Filter parameters (current defaults):*waitcycle =20, threshfactor=0.2, distfactor=0.75)*

Floudas Continuous Problems-results

- Best known solution found or improved on in 118 of 128 problems using base case parameters
- Success if “gap” < 1%
- Best OQNLP solution found on first or second Solver call (itn 201) in 99 of 118 solved problems
- All the 10 remaining problems are solved by either loosening the filters or using (1000,5000)
- Solver is called at only 1% to 6% of the trial points where a call is allowed.

Performance by # of Variables (avg)

variable range	problems	variables	constraints	solver calls	locals found	function calls	time to best	total time	failed	first *L* call	second *L* call
1 to 4	32	3	2	8	2	2158	0.1	0.5	1	27	3
4 to 7	31	6	6	6	2	4767	0.2	0.6	0	22	6
8 to 12	21	9	8	13	3	19698	0.1	0.8	3	10	4
13 to 20	18	16	12	7	3	5212	0.3	0.7	4	8	1
22 to 78	13	39	28	14	3	23078	0.6	2.5	1	5	3
110 to 141	13	116	80	24	23	NA	6.6	64.1	0	7	2
total/avg	128			11	4	7191	0.4	1.7	9	80	19

Solving 6 Lennard-Jones Problems Using CONOPT and Loose Filters

variables	constraints	Solver calls to best	Total Solver calls	locals found	time to best	total time	Gap,%
9	10	1	152	39	1.09	21.83	0.00
24	45	21	130	114	18.56	68.34	0.00
39	105	6	104	100	13.63	165.09	0.00
54	190	67	118	118	257.6	396.21	1.12
69	300	42	94	94	325.8	730.68	1.84
84	435	16	59	59	134.4	434.56	0.88

Comparison-OQNLP and Random Starts

- ***Floudas Series 8_6_1_x - Lennard-Jones Potential Function***

atoms	vars	con	calls	nglob	diff	exp calls	oqnlp calls
5	9	10	100	99	2	1.0	1
10	24	45	100	4	27	25.0	21
15	39	105	100	3	85	33.3	6
20	54	190	200	2	167	100	67
25	69	300					42
30	84	435					16

- nglob = number of times global min found
- diff = number of different local minima found
- pglob = nglob/calls
- exp calls = expected number of random starts until global first found
- = sum(0=<k<inf, k*pglob*(1-pglob)^(k-1)]=1/pglob
- oqnlp calls = number of OQNLP solver calls to find global min

Comparison-OQNLP and Random Starts

Floudas Series 8_6_2_x - Morse Potential Function

Atoms	vars	con	calls	nglob	diff	exp calls to best	oqnlp calls to best
5	9	0	200	13	17	15.4	1
10	24	0	200	1	107	200	1
15	39	0	200	9	161	22.2	1
20	54	0	200	10	189	20	2
25	69	0	200	2	185	100	4
30	84	0	200	2	188	100	17
40	114	0	200	3	191	66.7	7
50	144	0	200	3	181	66.7	20

Solving 13 MINLP's from Floudas set

- All are small—max binaries=8, max cont=23
- When OQ varies only binaries (subject to linear constraints
  - No stage one
  - All terminate after complete enumeration
  - Average(calls to best/total calls) approx 0.5
  - *L* called at every trial point
- When OQ varies all variables
  - Stage 1 helps find better values of binaries for stage 2
  - Average(calls to best all/calls to best bin only)=0.7

Advantages and Disadvantages

- Advantages
  - Finds good or optimal solutions rapidly
  - Can handle problems of general form with hundreds or thousands of continuous variables and/or constraints
  - strong ability to find a feasible solution (due to NLP solver)
  - works very differently from DICOPT or branch and bound
- Disadvantages
  - No guarantee of finding a global solution
  - No way to tell if you have found a global solution

Improving Reliability of NLP Solvers

- Many solver calls from diverse starting points are much more likely to succeed (find a local solution) than a single call
- OQNLP generates the starting points automatically and insures diversity
- OQNLP keeps track of all solutions found, returns best + all others if required.
- NLP solver failures are often due to termination at an infeasible point. Can recast problem as feasibility problem, apply OQNLP
- Feasibility mode: will eliminate need to reformulate problem (in future).

Improving Solver Reliability-Results

- Solved 147 Globallib problems with Conopt and OQNLP
- Conopt failed or ended infeasible in 24
- OQNLP got local solutions to 14 of these with 2 solver calls (58%)
- OQNLP got local solutions to 16 with 1000 iterations, about 40 solver calls per run avg.

GAMS and Callable Library Versions

- GAMS beta test release was 6/03 (see OR/MS today for Ad)
- Full user guide available
- Comparisons with Baron and LGO are planned in ‘03 using the test problems in Globalworld and Minlpworld.

- OQNLP Callable Library also available from OPTTEK Systems or Optimal Methods Inc

Future Work-feeding back local solutions to OQ

- Currently local solutions not returned to OQ
- Could start with initial NLP call at user initial point, pass NLP solution to OQ to include in initial population.
- In subsequent NLP calls, pass some NLP solutions back to OQ (must maintain diversity of its population).
- Criteria for which solutions are returned to OQ not clear.

More Future Work

- Determine effects of varying important OQMS options and parameters
  - basin and merit filter parameters
  - Performance of dynamic filters
  - OQ treats continuous vars as discrete
  - OQ strategies to intensify or diversify search
  - fractional change criterion for overall method
- Comparison with LGO,MLSL of Fylstra, DICOPT
- Improvements to OQMS algorithm motivated by test results

Better starting points for minlp

- If Optquest varies only the discrete variables, can generate all trial points arising from a given reference set at once
- Choose a "best" point (minimal penval with continuous variables zero), sort the remaining points in increasing order of their distance from the "best" one.
- Solve the NLP subproblems in this order, using the previous optimal solution as the initial values for the continuous variables.
- Effects are being measured, but larger test problems probably needed.