CONSTRAINED GENETIC ALGORITHMS AND THEIR APPLICATIONS IN NONLINEAR CONSTRAINED OPTIMIZATION

Benjamin W.Wah and Yixin Chen

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{wah,chen}@manip.crhc.uiuc.edu

ICTAI 2000, USA

November 14, 2000

Outline

- Discrete Constrained Nonlinear Programming Problems (NLPs)
- Previous Work
  - Discrete Constrained Optimization using Lagrange Multipliers
  - Genetic Algorithms for Constrained Optimization
- A General Framework to Look for Discrete-Space Saddle Points
  - Constrained Genetic Algorithms using Lagrange Multipliers
  - Optimal CGA with Iterative Deepening
- Experimental Results
- Conclusions

Application-Problem Definition

- A general discrete constrained NLP:

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & g(x) \leq 0 & x = (x_1, x_2, \ldots, x_n) \text{ is a vector} \\ & h(x) = 0 & \text{of finite discrete variables,} \end{array}$$

  - $f(x)$ is a lower-bounded objective function
  - $h(x) = [h_1(x), \cdots, h_m(x)]^T$ is a vector of m equality constraints
  - $g(x) = [g_1(x), \cdots, g_k(x)]^T$ is a vector of k inequality constraints

Discrete Constrained Optimization Using Lagrange Multipliers

- Generalized augmented Lagrangian function with discrete x

$$L_d(x, \lambda) = f(x) + \lambda^T H(h(x)) + \frac{1}{2}||h(x)||^2.$$

- Discrete-neighborhood saddle point (SP_{dn}) (x^*, λ^*):

$$L_d(x^*, \lambda) \leq L_d(x^*, \lambda^*) \leq L_d(x, \lambda^*)$$

  for all $x \in \mathcal{N}_{dn}(x^*)$ (discrete neighborhood of x^*) and all $\lambda \in R^{n+m}$.

  - L_d is at local maximum in λ space (implying constraint satisfaction)
  - L_d is at local minimum in x space (implying local minimum in $f(x)$)

- Theorem: First-order necessary and sufficient condition
  - If all constraint functions are transformed to be non-negative then there is a one-to-one correspondence between CLM_{dn} (constrained local minimum under discrete neighborhoods) and SP_{dn}

Discrete Lagrangian Method (DLM)

- Given a user-defined neighborhood $\mathcal{N}(\mathbf{x})$, $\lambda_0 \leftarrow 0$, and $\mathbf{x}_0 = (x_0, \lambda_0)$
  - Generate a trial point $\mathbf{x}'$ from $\mathcal{N}(\mathbf{x})$ using $G(\mathbf{x}, \mathbf{x}')$,
  - Accept $\mathbf{x}'$ if L_d is reduced wrt x or increased wrt λ
  - Stop search when no further improvements can be found
- If user-defined neighborhood is small enough to be enumerated in each iteration, then search can find discrete-neighborhood CLM

Constrained Simulated Annealing (CSA)

- Given a user-defined neighborhood $\mathcal{N}(\mathbf{x})$, $\lambda_0 \leftarrow 0$, $\mathbf{x}_0 = (x_0, \lambda_0)$,

  T_0: initial temperature,

  N_T: number of trials per temperature,

  α: cooling rate ($0 < \alpha < 1$

  - Generate a trial point $\mathbf{x}'$ from $\mathcal{N}(\mathbf{x})$ using $G(\mathbf{x}, x')$,
  - Accept $\mathbf{x}'$ if L_d using Metropolis probability $A_T(\mathbf{x}, x')$ if L_d is reduced wrt x or increased wrt λ
  - Reduce T by α and repeat steps until $T_i \leq T_\infty$

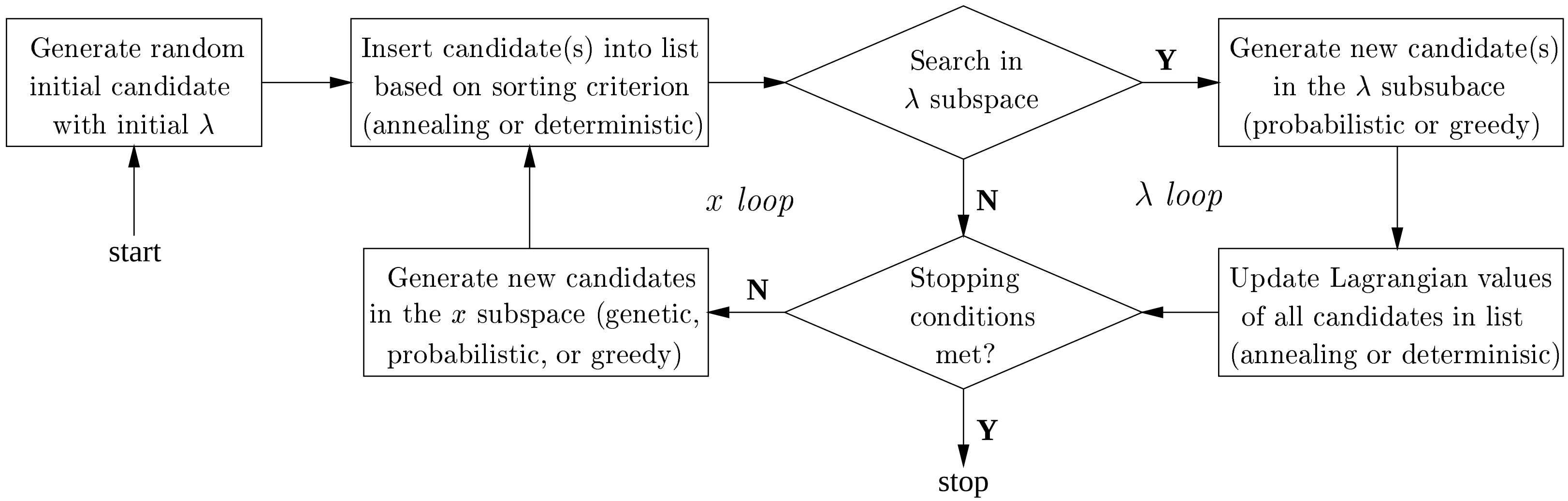
- Asymptotic Convergence Theorem (ICTAI'99)
  - Markov chain modeling CSA converges asymptotically to a constrained global minimum (CGM_{dn}) with probability one

Existing Genetic Algorithms for Constrained Optimization

- Penalty Methods: static, dynamic, and annealing penalties, $\cdots$
- Domain-specific knowledge or problem-dependent genetic operators
  - Examples: preserving feasibility, decoder-based methods, repair methods, co-evolutionary methods, strategic oscillation, $\cdots$
  - Problem-dependent performance that requires extensive tuning
- Local minima in penalty functions are only *necessary but not sufficient* to be CLM in the original constrained NLP, unless penalties are chosen suitably
  - Difficult to choose suitable penalties

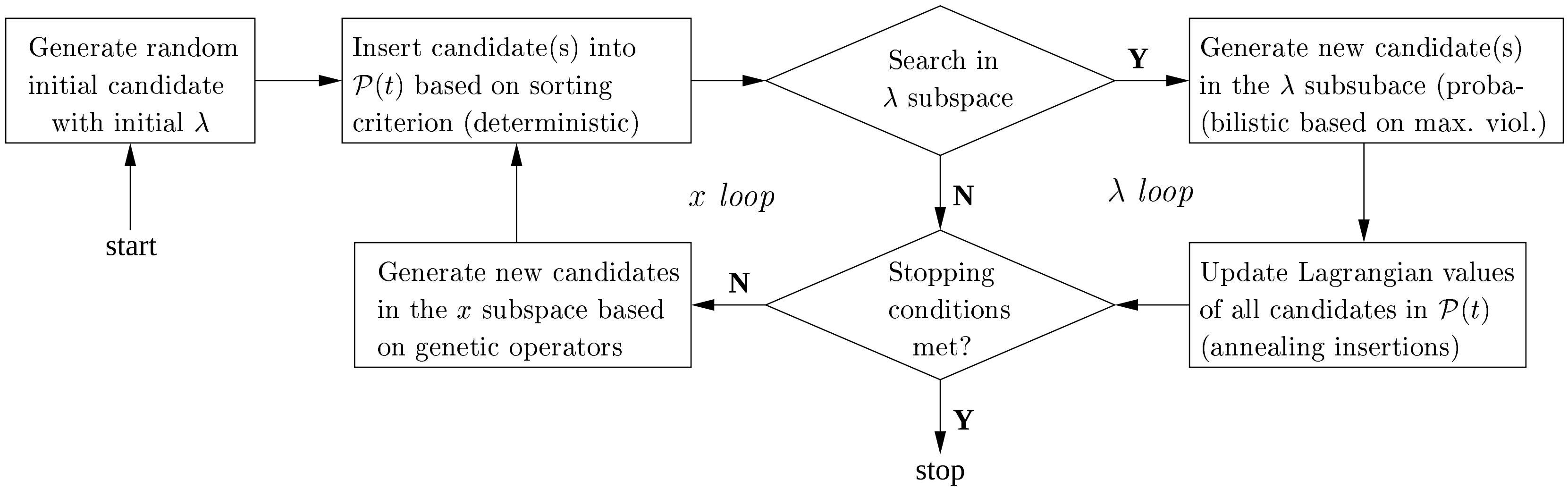
CONSTRAINED GENETIC ALGORITHMS

General Framework to Look for Discrete-Space Saddle Points



- Stop at either feasible points or at CLM if all $x \in \mathcal{N}_d(x)$ can be enumerated
- Both DLM and CSA fit into this framework
- CGA: Extend the mechanisms by including genetic operators

Constrained Genetic Algorithm (CGA)



- Descents in original X subspace using genetic algorithm
- Using Lagrangian function as the fitness function
- Necessary condition for CGA to converge
  - All individuals are feasible solutions to the original problem
- Difficulties: Determine proper population size and number of generations

Combined Constrained SA and GA (CSAGA)

1. **procedure** CSAGA(P, N_g)
2. set $t \leftarrow 0$, T_0, $0 < \alpha < 1$, and $\mathcal{P}(t)$;
3. **repeat** /* over multiple generations */
4. **for** $i \leftarrow 1$ **to** P **do** /* SA in Lines 5-11 */
5. **for** $j \leftarrow 1$ **to** $freq$ **do**
6. generate $\mathbf{x}'_j$ from $\mathcal{N}_d(\mathbf{x_j})$ using $G(\mathbf{x_j}, \mathbf{x'_j})$;
7. accept $\mathbf{x'_j}$ with probability $A_T(\mathbf{x_j}, \mathbf{x'_j})$
8. **end_for**
9. **end_for**
10. set $T \longleftarrow \alpha \times T$; /* set T for the SA part */
11. **repeat** /* by GA over probes in x subspace */
12. $y \leftarrow GA(select(\mathcal{P}(t)))$;
13. evaluate $L_d(y, \lambda)$ and insert y into $\mathcal{P}(t)$;
14. **until** sufficient number of probes in x subspace;
15. $t \leftarrow t + freq$; /* update generation number */
16. **until** $(t \geq N_g)$
17. **end_procedure**

OPTIMAL CGA WITH ITERATIVE DEEPENING

Reachability Probabilities in Stochastic Searches

- A single CGA run with N_g generations and population size P succeeds in finding a solution with *reachability probability* $P_R(N_g, P)$
- CGA with finite N_g can be run multiple times to improve success prob.
  - B_{N_g}: total number of probes in multiple independent runs of CGA (each with N_g generations) to find a solution

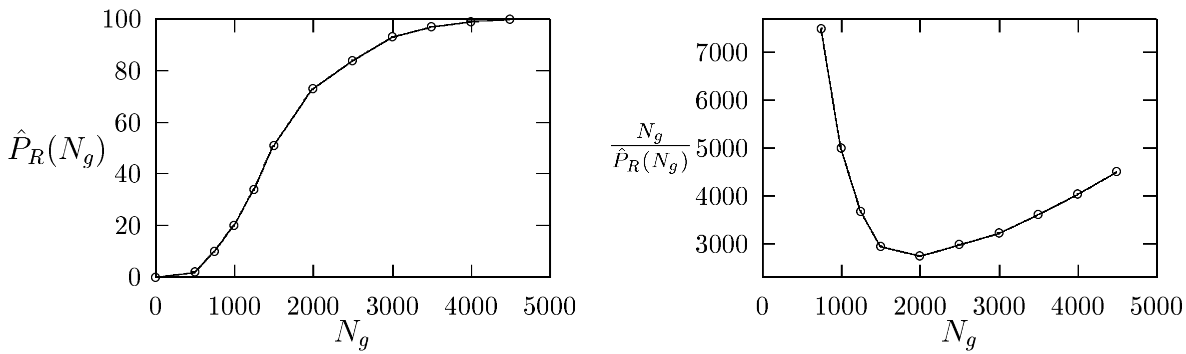
$$E(B_{N_g}|P) = \sum_{j=1}^{\infty} P_R(N_g, P)(1 - P_R(N_g, P))^{j-1} N_g \times P \times j$$
$$= \frac{N_g \times P}{P_R(N_g, P|P)} = P \frac{N_g}{\hat{P}_R(N_g)}$$

where $\hat{P}_R(N_g)$ is $P_R(N_g, P|P)$ and $\frac{N_g}{\hat{P}_R(N_g)}$ is generally convex wrt $\log(N_g)$

- Also applicable to CSAGA

Illustration: Convex Relationship in $E(B_{N_g})$

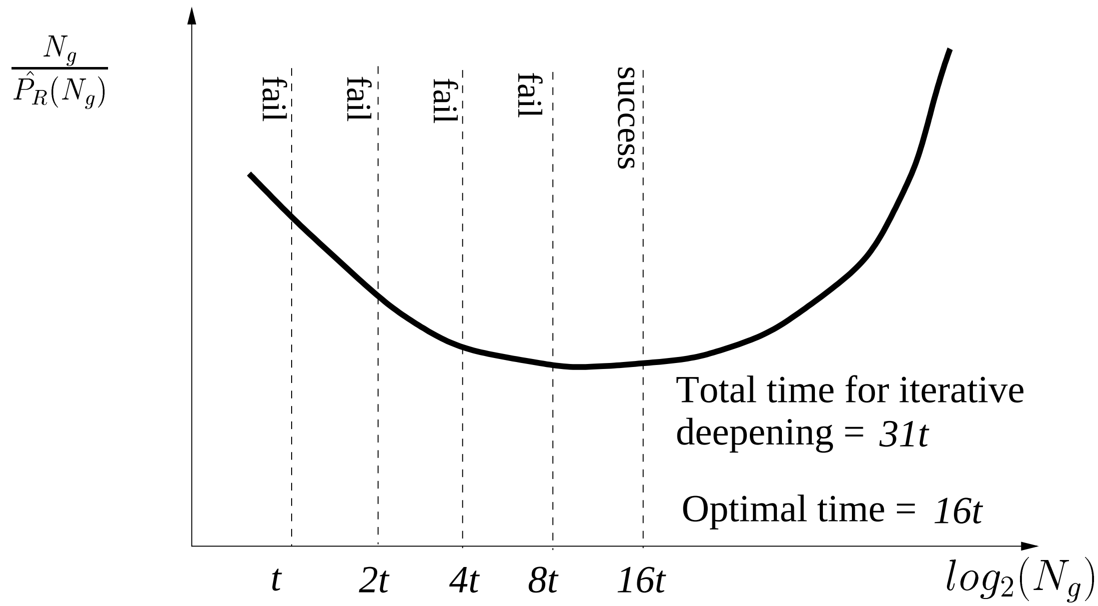
- Existence of optimal number of generations
  - Applying CSAGA with $P = 3$ to solve G1 ($N_{(g_{opt})} \approx 2000$)



a) $\hat{P}_R(N_g)$ satisfies the two sufficient conditions b) Absolute minimum in $\frac{N_g}{\hat{P}_R(N_g)}$

- Goal: design a strategy of running CGA/CSAGA to minimize the expected overhead of finding a solution
  - Prove that the average completion time of the new strategy is optimal (up to a constant factor)

CGA/CSAGA with Iterative Deepening



Procedure $CGA_{ID}/CSAGA_{ID}$

1. Set small N_g;
2. Execute CGA/CSAGA K times with N_g; If success, exit;
3. Double N_g; Goto Step 2;

Optimal Number of Generations

Sufficient conditions for existence of absolute minimum of $\frac{N_g}{\hat{P}_R(N_g)}$ versus N_g curve in $(0, +\infty)$

1. $\hat{P}_R(0) = 0$ and $\lim_{N_g \to \infty} \hat{P}_R(N_g) \leq 1$
2. $\hat{P}''_R(0) > 0$

Optimality of $CGA_{ID}/CSAGA_{ID}$

- $\hat{P}_R(N_g)$ is monotonically non-decreasing for N_g in $(0, \infty)$;
- $\frac{N_g}{\hat{P}_R(N_g)}$ versus N_g curve satisfies sufficient conditions for absolute minimum;
- $2 \times (1 - \hat{P}_R(N_{g_{opt}}))^K < 1 \Longrightarrow K = 3$ in procedure.

EXPERIMENTAL RESULTS

Experimental Results on G1-G10

Problem ID	Global Solution f^*	EAs Best Sol.	EAs Method	CSA_{ID} $\bar{\mathcal{T}}(f^*)$	CGA_{ID} P_{opt}	CGA_{ID} $\bar{\mathcal{T}}(f^*)$	$CSAGA_{ID}$ P	$CSAGA_{ID}$ $\bar{\mathcal{T}}(f^*)$	$CSAGA_{ID}$ P_{opt}	$CSAGA_{ID}$ $\bar{\mathcal{T}}(f^*)$
G1 (min)	-15	-15	Genocop	1.65	40	5.49	3	1.64	2	1.31
G2 (max)	-0.80362	0.803553	S.T.	7.28	30	311.98	3	5.18	3	5.18
G3 (max)	1.0	0.999866	S.T.	1.07	30	14.17	3	0.89	3	0.89
G4 (min)	-30665.5	-30664.5	H.M.	0.76	5	3.95	3	0.95	3	0.95
G5 (min)	4221.9	5126.498	D.P.	2.88	30	68.9	3	2.76	2	2.08
G6 (min)	-6961.81	-6961.81	Genocop	0.99	4	7.62	3	0.91	2	0.73
G7 (min)	24.3062	24.62	H.M.	6.51	30	31.60	3	4.60	4	4.07
G8 (max)	0.095825	0.095825	H.M.	0.11	30	0.31	3	0.13	4	0.10
G9 (min)	680.63	680.64	Genocop	0.74	30	5.67	3	0.57	3	0.57
G10 (min)	7049.33	7147.9	H.M.	3.29	30	82.32	3	3.36	3	3.36

- Algorithms derived from the framework can find optimal solutions without problem-dependent strategy and tuning used in EA
- CGA_{ID} is not competitive as compared to CSA_{ID} and $CSAGA_{ID}$
- Fixed $P = 3$ in $CSAGA_{ID}$ leads to minor performance loss

Experimental Results on Floudas and Pardalos Problems

- Selected large ($n_v > 10$) problems, $CSAGA_{ID}$ with fixed $P = 3$

Problem		$f(x)$	CSA_{ID}	$CSAGA_{ID}$
ID	Best Sol.	n_v	$\bar{\mathcal{T}}(f^*)$	$\bar{\mathcal{T}}(f^*)$
2.7.1(min)	-394.75	20	35.11	14.86
2.7.2(min)	-884.75	20	53.92	15.54
2.7.3(min)	-8695.0	20	34.22	22.52
2.7.4(min)	-754.75	20	36.70	16.20
2.7.5(min)	-4150.4	20	89.15	23.46
5.2(min)	1.567	46	3168.29	408.69
5.4(min)	1.86	32	2629.52	100.66
7.2(min)	1.0	16	824.45	368.72
7.3(min)	1.0	27	2323.44	1785.14
7.4(min)	1.0	38	951.33	487.13

- Substantial improvement:
  - Average time $CSAGA_{ID}$ takes to find the optimal solution is 1.3 to 26.3 times less than that of CSA_{ID}

Conclusions

- A general framework that unifies various approaches to look for discrete-space saddle points
  - Combined CSA and CGA leads to improved performance
- Iterative deepening: a general technique for determining the optimal schedules of stochastic algorithms at run time
  - Overhead of a (small) constant factor over the optimal completion time