Constrained Global Optimization by Constraint Partitioning and Simulated Annealing

Benjamin W. Wah
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
URL: http://manip.crhc.uiuc.edu

Yixin Chen and Andrew Wan
Department of Computer Science and Engineering
Washington University
St. Louis, MO 63130

Outline

- Key observation
  - Constraints in many application problems are structured
- Partition-and-resolve approach
  - Partition a problem by its constraints into subproblems
- Simulated annealing
  - Constrained simulated annealing
  - Constrained partitioned simulated annealing
- Experimental results
- Conclusions

Nonlinear Constrained Optimization

- An application problem defined by
  - A set of mixed (discrete and continuous) variables
  - A nonlinear objective function
  - A set of nonlinear constraints (conditions to be satisfied in the application)
- Exists in every engineering field
  - Planning of spacecraft and satellite operations
  - Placement and routing of components in a VLSI chip
  - Design of aircrafts
  - Design of a petroleum pipeline

Example MINLP Trimlon12

- TRIMLON
  - Minimize the trim loss in producing a set of paper rolls from raw paper rolls
  - Trimlon12 has 168 variables (integer *n*, real *y*, *m*) and 72 constraints
    - Not solvable by any existing MINLP solver from the starting point specified

variables: $y[j],\ m[j],\ n[j,i]$ where $i = 1, \cdots, I;\ j = 1, \cdots, J$

objective: $\min_{z=(y,m,n)} f(z) = \sum_{j=1}^{J} (c[j] \cdot m[j] + C[j] \cdot y[j])$ (OBJ)

subject to:

$$B_{min} \le \sum_{i=1}^{I} (b[i] \cdot n[i,j]) \le B_{max} \quad \text{(C1)}$$

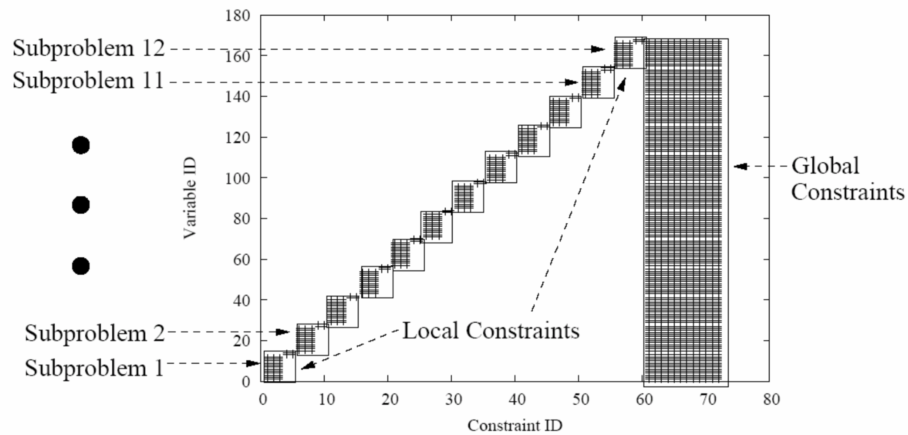
$$\sum_{i=1}^{I} n[i,j] - N_{max} \le 0 \quad \text{(C2)}$$

$$y[i] - m[j] \le 0 \quad \text{(C3)}$$

$$m[j] - M \cdot y[j] \le 0 \quad \text{(C4)}$$

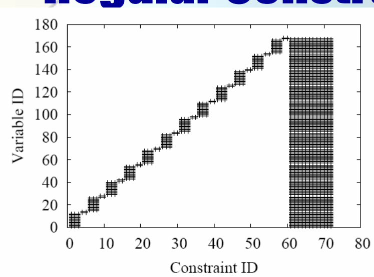
$$Nord[i] - \sum_{j=1}^{J} (m[j] \cdot n[i,j]) \le 0. \quad \text{(C5)}$$

Constraint Locality in TRIMLON12 [Wah05]

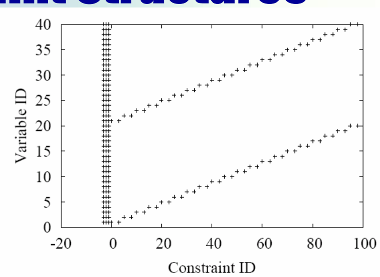


12 out of 72 constraints (16.7%) are global constraints

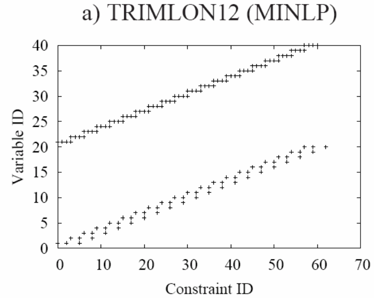
Regular Constraint Structures



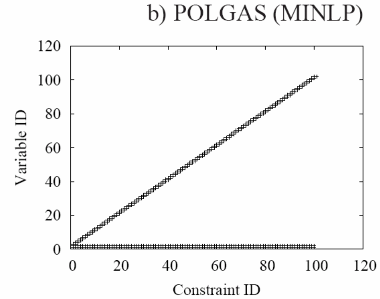
a) TRIMLON12 (MINLP)



b) POLGAS (MINLP)



c) OPTCDEG3 (CNLP)



d) ORTHRGDS (CNLP)

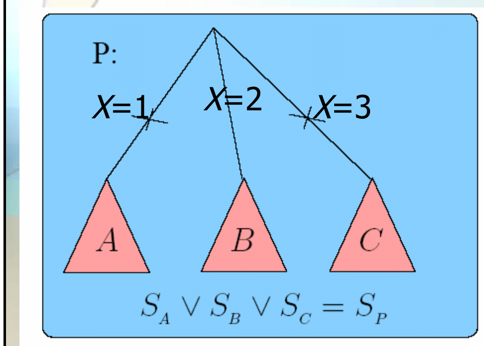
Real Constraints Are Structured

- Constraints model entities and actions with spatial or temporal locality
  - Relations among components close to each other in space for problems of physical structures
  - Relations among actions close to each other in time for scheduling problems
- A majority of the application problems encountered have structured constraints

Outline

- Key observation
  - Constraints in many application problems are structured
- **Partition-and-resolve approach**
  - **Partition a problem by its constraints into subproblems**
- Simulated annealing
  - Constrained simulated annealing
  - Constrained partitioned simulated annealing
- Experimental results
- Conclusions

Subspace Partitioning



Subspace Partitioning

Partition **P by branching on the values of a variable**

Solve P by choosing the correct path and by solving the subproblem

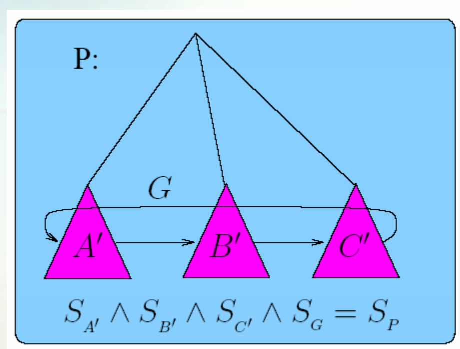
Overhead for solving each subproblem *is similar* to that of P

Constraint Partitioning

Partition **P by its constraints into subproblems**

Solve P by solving all the subproblems and by resolving those violated (active) global constraints

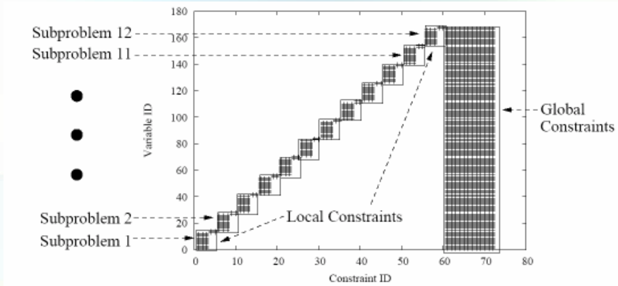
Overhead of each subproblem is substantially smaller



Constraint Partitioning

Each subproblem is significantly relaxed with a much larger solution space

Illustration on Constraint Partitioning



- Subproblem
  - Subproblem of satisfying the order from one customer
  - Local constraints: requirements from one customer
  - Subproblem substantially easier to solve than original problem
  - Each subproblem can be solved by the solver of the original problem
- Very few global constraints
  - All the orders must fit into the single paper roll produced

Previous Work: Penalty Methods

- General penalty formulation

$$L(variable, penalty) = objective + penalty \sum constraint\ violations$$

  - When the penalty is large enough
    - Global minimum of penalty function (hard to find) $\Leftrightarrow$ constrained global minimum of the original problem
- KKT
  - Solving a system of nonlinear equations (easier) for some exact penalties of the penalty function

  Local minimum of penalty function $\Rightarrow$ constrained local minimum of original problem
  - Differentiability and continuity requirements
  - Process cannot be partitioned

Theory of Extended Saddle Points [WC06]

- When penalties > some threshold (easy)

  Extended saddle point of penalty function

  ⇔ Constrained local minimum of original problem

  - Loose assumptions, without continuity and differentiability of constraint functions
- Partitioning of the N&S condition into a set of necessary conditions that are sufficient collectively
  - One necessary condition for each subproblem
  - One necessary condition for the global constraints

ILLINOIS

Naïve Partition-and-Resolve Framework

$$\min_{z} J(z)$$

$$\text{subject to } h^{(t)}(z(t)) = 0, \quad g^{(t)}(z(t)) \le 0 \quad \text{(local constraints)}$$

$$H(z) = 0, \quad G(z) \le 0 \quad \text{(global constraints)}$$

$$H(z) = 0 \text{ and } G(z) \le 0$$

$$\min_{z(1)} J(z) \quad \text{s.t. } h^{(1)}(z(1) = 0, \quad g^{(1)}(z(1)) \le 0$$

• • •

$$\min_{z(n)} J(z) \quad \text{s.t. } h^{(n)}(z(n) = 0, \quad g^{(n)}(z(n)) \le 0$$

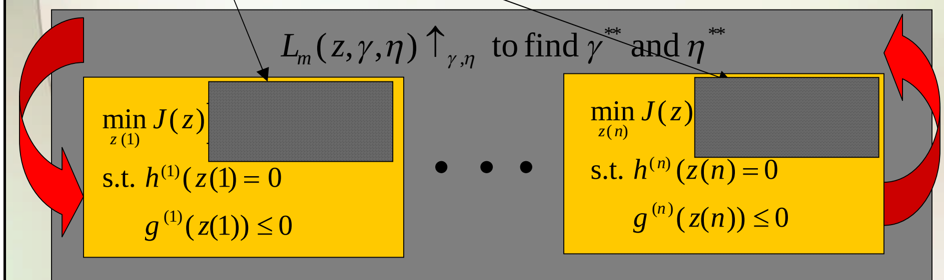
ILLINOIS

Partition-and-Resolve Framework [WC06]

Weighted active global constraints provide guidance in local subproblems

Similar solver as original problem

- Solving a subproblem
  - Satisfy local constraints
  - Minimize global objective
  - Satisfy (soft) global constraints
- Increasing penalties on violated global constraints



Outline

- Key observation
  - Constraints in many application problems are structured
- Partition-and-resolve approach
  - Partition a problem by its constraints into subproblems
- Simulated annealing
  - Constrained simulated annealing
  - Constrained partitioned simulated annealing
- Experimental results
- Conclusions

Simulated Annealing [Kirkpatrick83]

- Unconstrained optimization problem $\min_z f(z)$
- Accept probes by an acceptance probability governed by a Metropolis probability

$$A_T(z, z') = e^{\left(-\frac{\max(f(z') - f(z), 0)}{T}\right)}$$

- Converge asymptotically to a globally optimal solution with probability one
  - Discrete optimization problem
  - Properly chosen temperature schedule

ILLINOIS

Constrained Simulated Annealing [WW99]

- Constrained optimization problem
- Descend in *z* subspace and ascend in α subspace
- Acceptance probability governed by temperature that is reduced by a cooling schedule

$$A_T(z, z') = \begin{cases} e^{\left(-\frac{\max(L(z',\alpha) - L(z,\alpha), 0)}{T}\right)} & \text{descend in } z \text{ subspace} \\ e^{\left(-\frac{\max(L(z,\alpha) - L(z,\alpha'), 0)}{T}\right)} & \text{ascend in } \alpha \text{ subspace} \end{cases}$$

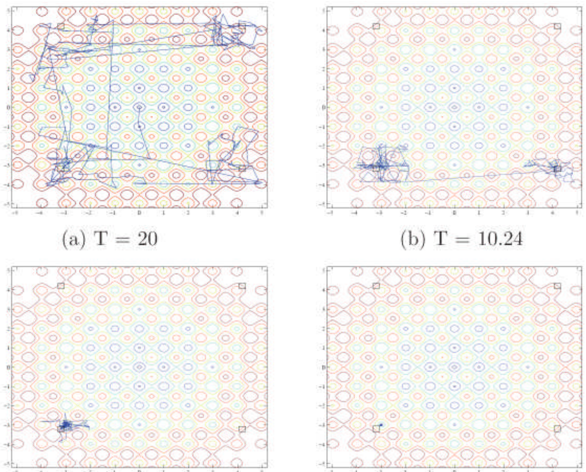
- Converge asymptotically to a constrained global minimum with probability one [WCW07]
  - Discrete optimization problem
  - Properly chosen temperature schedule

ILLINOIS

Illustration

$$\min_{x_1, x_2} \quad f(x) = 10n + \sum_{i=1}^{2}\left(x_i^2 - 10\ \cos(2\pi x_i)\right) \quad \text{where } x = (x_1, x_2)^T$$

$$\text{subject to} \quad |(x_i - 3.2)(x_i + 3.2)| = 0, \quad i = 1, 2.$$



(a) T = 20 (b) T = 10.24

(c) T = 8.192 (d) T = 0.45

Outline

- Key observation
  - Constraints in many application problems are structured
- Partition-and-resolve approach
  - Partition a problem by its constraints into subproblems
- Simulated annealing
  - Constrained simulated annealing
  - Constrained partitioned simulated annealing
- **Experimental results**
- Conclusions

Experimental Results

- CUTE nonlinear optimization benchmarks
- Compare CSA, CPSA, GEM, two other existing penalty methods
- CSA
  - One order of magnitude faster
  - Find better solutions
- CPSA
  - Two order of magnitude faster
  - Find much better solutions

Conclusions

- Constraint partitioning is a powerful approach for exploiting constraint structure in order to reduce complexity
  - Bottom-up resolution with guidance provided by top-level active global constraints
  - Using existing solvers to solve partitioned subproblems
- CPSA
  - Overcome the complexity limitations of CSA
  - Find much better solutions than CSA
  - Establish the theoretical foundation of the constraint partitioning approach