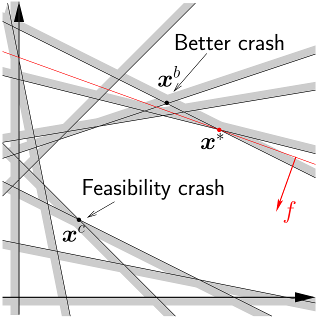
Overview

A linear programming problem is

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$



$$\mathbf{x}^0 \xrightarrow{\text{Crash}} \mathbf{x}^k \xrightarrow{\text{Crossover}} \mathbf{x}^b \xrightarrow{\text{Simplex}} \mathbf{x}^*$$

Motivation

- A good starting basis is essential for the solution of large problems
- The Idiot crash is a heuristic aiming to improve feasibility
- Implemented by John Forrest in CLP
- Seeking primal feasibility prior to primal simplex

$$\mathbf{x}^0 \xrightarrow{\text{Crash}} \mathbf{x}^k \xrightarrow{\text{Crossover}} \mathbf{x}^b \xrightarrow{\text{Simplex}} \mathbf{x}^*$$

Idiot is Augmented Lagrangian

- No documentation but the code
  - John Forrest: "so bad that I called it the Idiot algorithm"
- At each iteration minimize a weighted average of objective and infeasibilities
- Augmented Lagrangian (AL)
- Aspects that make it different from AL
  - Approximate minimization
  - Parameter adjustment
  - Other details

Augmented Lagrangian

- In iteration k

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} + \sum_{i \in 1..m} \sigma_i^k (\mathbf{r}_i(\mathbf{x}^k) + \theta_i^k)^2,$$

- σ_i and θ_i are parameters corresponding to the i^{th} row
- $r_i(\mathbf{x}^k)$ is the residual of the i^{th} row
- Adjust parameters

At start	After some iterations s	After some iterations l
$\boldsymbol{\sigma}^0 = \mathbf{1}$		$\boldsymbol{\sigma}^{l+1} := 10\boldsymbol{\sigma}^l$
$\boldsymbol{\theta}^0 = \mathbf{0}$	$\boldsymbol{\theta}^{s+1} := \boldsymbol{\theta}^s + \mathbf{r}(\mathbf{x}^s)$	$\boldsymbol{\theta}^{l+1} := \boldsymbol{\theta}^l/10$

Alternating directions approach

- Similar to Alternating Direction Method of Multipliers (ADMM)

$$\min_{\mathbf{x}} h(\mathbf{x}) \quad \Bigg| \quad \begin{array}{c} \min_{x_1} h(\mathbf{x}) \\ \min_{x_2} h(\mathbf{x}) \\ \min_{x_3} h(\mathbf{x}) \\ \dots \\ \min_{x_n} h(\mathbf{x}) \end{array}$$

- Approximate solution
- repeated ≈ 100 times to improve accuracy

The Idiot Crash

The Idiot Crash structure

Initialize $\mathbf{x}^0$, σ^0, $\boldsymbol{\theta}^0 = \mathbf{0}$

For $l \in 1..\mathrm{nMajor}$

For $i \in 1..\mathrm{nMinor}$

$$\forall j \in \{1..n\} \quad \min_{x_j} \quad \mathbf{c}^T\mathbf{x} + \sum_{i \in 1..m} \sigma^k (\mathbf{r}_i(\mathbf{x}^k) + \theta_i^k)^2$$

end

Choose new σ or $\boldsymbol{\theta}$

end

- $\sigma^k = \sigma_1^k = \sigma_2^k = .. = \sigma_m^k$
- $\theta_i^{s+1} := \frac{1}{2} r_i(\mathbf{x}^s)$

Parameters

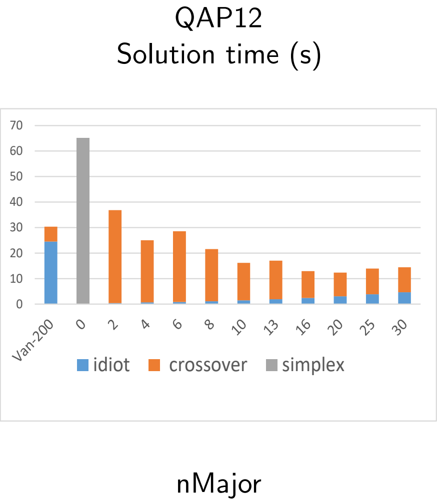
At the end of each major iteration the values of $\boldsymbol{\theta}$ or σ are re-adjusted:

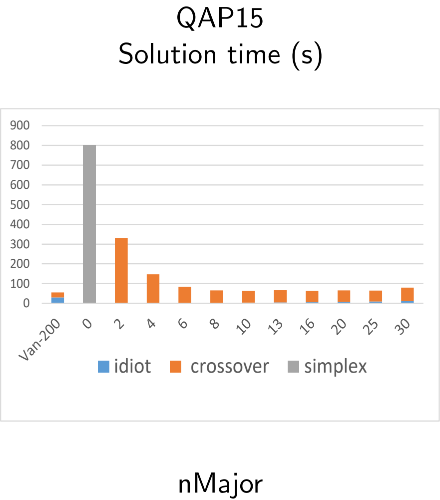
- Start with σ^0, $\boldsymbol{\theta}^0 = \mathbf{0}$, $\text{nMinor} = 2$
- At the end of each major iteration:
  - Until 10% reduction in primal infeasibility set $\sigma^{k+1} = 10\sigma^k$
- Set $\text{nMinor} = 105$
  - The first major iteration with a new σ, $\boldsymbol{\theta}$ is unchanged
  - The next few major iterations set $\theta_i^{k+1} = \frac{1}{2} r_i(\mathbf{x}^k)$
  - Increase σ and repeat

Performance

- Generally useless
  - Single variable change limitation
- Solves Quadratic Assignment Problems (QAPs)
  - Combinatorial optimization problem, special case of facility location
  - Well known for being hard to solve
  - The Idiot crash reduces solution time significantly

Quadratic Assignment Problems





Idiot alone

$$\begin{aligned} \min \quad & f_P = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \qquad \text{(P)} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \max \quad & f_D = \mathbf{b}^T\mathbf{y} \\ \text{s.t.} \quad & A^T\mathbf{y} \leq \mathbf{c} \qquad \text{(D)} \end{aligned}$$

Idiot applied to (P)

- Primal feasible
- Near-optimal objective $f_P \geq f^*$

Idiot applied to (D)

- Dual feasible
- Objective $f_D \leq f^*$

Observations

				Time (s)		
Problem	LB	UB	f^*	Bounds	Idiot+Crossover	Simplex
qap08	141.39	204.57	203.50	0.55	0.61	1.80
qap12	481.50	523.60	522.89	3.85	12.64	65.21
qap15	950.05	1042.05	1040.99	11.53	64.48	803.30

- QAPs require integer solution to linearization
- Fast bounds on f^* accelerate branch & bound algorithms