Crash-Starting the Simplex Method

Ivet Galabova   Julian Hall

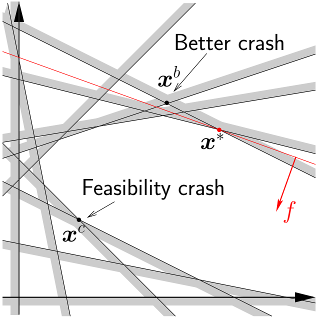
School of Mathematics, University of Edinburgh

Optimization Methods and Software

December 2017

A linear programming problem is

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



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

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

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

The Idiot crash

- 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
- Looks like quadratic penalty function approach but...
  - Extra term in function minimized
- Looks like Augmented Lagrangian (AL) but...
  - Approximate minimization
  - Parameter adjustment
  - Other details

Augmented Lagrangian

- In iteration k

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}_A = \mathbf{c}^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{r}(\mathbf{x}) + \frac{\mu}{2}\mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x}),$$

- Residual vector $\mathbf{r}(\mathbf{x}) = \mathbf{b} - A\mathbf{x}$
- Parameters μ and $\boldsymbol{\lambda}$
  - λ_i corresponds to the i^{th} row
  - $\boldsymbol{\lambda}$ converge to the optimal Lagrange multipliers
- Adjust parameters

At start	After some iterations s	After some iterations l
$\mu^0 = 2$		$\mu^{l+1} := 10\mu^l$
$\boldsymbol{\lambda}^0 = \mathbf{0}$	$\boldsymbol{\lambda}^{s+1} := \boldsymbol{\lambda}^s - \mu\mathbf{r}(\mathbf{x}^s)$	

Alternating variables approach

- $\mathcal{L}_A = \mathbf{c}^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{r}(\mathbf{x}) + \frac{\mu}{2}\mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x})$
    - Singular Hessian
    - Computationally expensive
- Alternating variables approach

$\min_{\mathbf{x}} h(\mathbf{x})$	$\min_{x_1} h(\mathbf{x})$ $\min_{x_2} h(\mathbf{x})$ $\dots$ $\min_{x_n} h(\mathbf{x})$

- Approximate solution
    - Cheap
- repeated ≈ 100 times to improve accuracy

The Idiot Crash structure

Initialize $\mathbf{x}^0$, μ^0, $\boldsymbol{\lambda}^0 = \mathbf{0}$
For $l \in 1..\mathrm{nMajor}$
  For $i \in 1..\mathrm{nMinor}$
    $$\forall j \in \{1..n\} \quad \min_{x_j} h(\mathbf{x}) = \mathbf{c}^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{r}(\mathbf{x}) + \frac{\mu}{2}\mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x})$$
  end
  Choose new μ or $\boldsymbol{\lambda}$
end

Every few major iterations μ is increased

Parameters

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

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

Idiot nature

- Not Augmented Lagrangian
- During initial major iterations μ is smaller so $\boldsymbol{\lambda}$ ensures reducing primal infeasibility has enough weight

$$\min_{x_j} h(\mathbf{x}) = \mathbf{c}^T\mathbf{x} - \boldsymbol{\lambda}^T\mathbf{r}(\mathbf{x}) + \frac{\mu}{2}\mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x}),$$

- Cycles through all columns
  - Random starting point
  - Alternating variables method
- Early termination of a major iteration is possible
  - Condition on moving average of expected progress

Relevant work

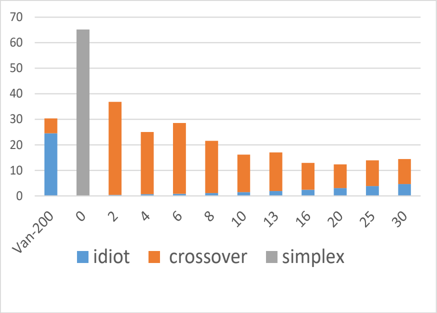
- **An Augmented Lagrangian Method for Linear Programming**
  Beale, Hattersley, James (1985)
- **The LP dual active set algorithm**
  Hager (1998)
- **An approximate, efficient LP solver for LP rounding**
  Sridhar, Wright et al (2013)
- **Sparse linear programming via primal and dual augmented coordinate descent**
  Yen, Zhong et al (2015)

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 linearisation solution time significantly
  - Many zero Lagrange multipliers

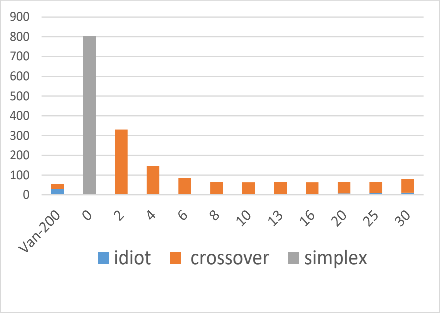
Quadratic Assignment Problems

QAP12
Solution time (s)



nMajor

QAP15
Solution time (s)



nMajor

Accuracy

- Residual error $\approx 10^{-4}$
  The Idiot crash on QAPs
- Residual error $\approx 10^{-1}$
  An approximate, efficient LP solver for LP rounding
  Sridhar, Wright et al (2013)
- Residual error $\approx 10^{-3}$
  Sparse linear programming via primal and dual augmented coordinate descent
  Yen, Zhong et al (2015)

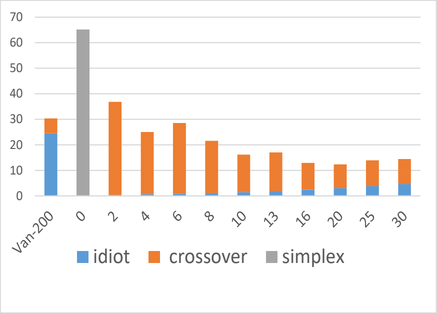
Change in objective value

Iteration	Penalty	Idiot	AL
0	0	0	0
1	449.307	449.307	0
2	447.728	447.672	4.759
3	446.731	446.676	0
4	520.996	520.936	236.608
5	519.322	519.324	110.543
6	518.447	518.448	492.086
7	527.641	527.642	457.191
9	527.391	527.393	530.405

Penalty method vs Idiot vs Augmented Lagrangian objective value

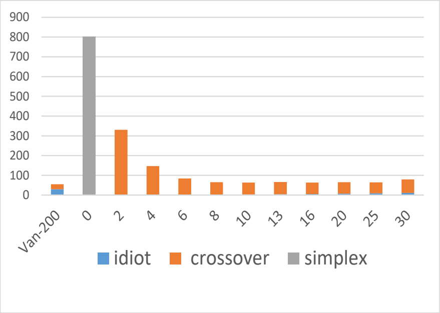
Quadratic Assignment Problems

QAP12
Solution time (s)



nMajor

QAP15
Solution time (s)



nMajor

Idiot alone

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

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

Idiot applied to (P)

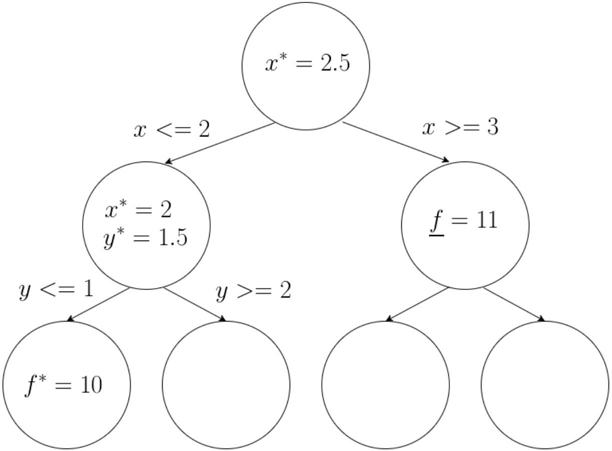
- Primal feasible
- Objective $f_P \geq f^*$
- Near-optimal, but not provably

Idiot applied to (D)

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

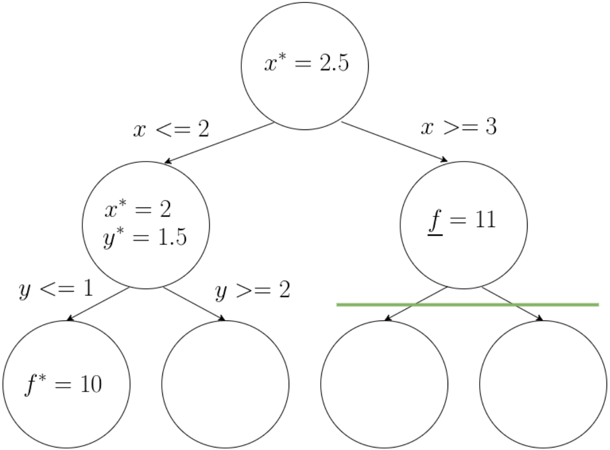
Branch & bound

- Branch & bound tree
- LP problems at each node
- Lower bounds on f^*
  - Allow for pruning



Branch & bound

- Branch & bound tree
- LP problems at each node
- Lower bounds on f^*
  - Allow for pruning



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

Conclusion

- Finding a good starting basis beneficial but difficult
- The Idiot crash is somewhere between a quadratic penalty function method and an Augmented Lagrangian method
- Limitations
  - Sensitive to parameters
  - Can have disastrous effects
  - Single variable change
- Effective on some problems
- Scope for accelerating B&B