

A quadratic penalty algorithm for linear programming and its application to linearizations of quadratic assignment problems

Julian Hall Ivet Galabova

School of Mathematics
University of Edinburgh

Computational Optimization in Action

8 June 2018



THE UNIVERSITY
of EDINBURGH

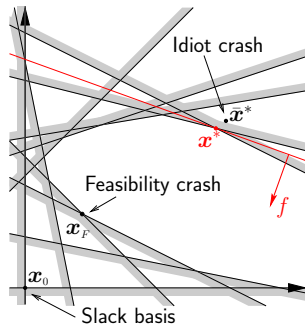


Solving LP problems: Crash start

$$\text{minimize } f = \mathbf{c}^T \mathbf{x} \quad \text{subject to } \mathbf{Ax} \leq \mathbf{b} \quad \mathbf{x} \geq \mathbf{0}$$

Choosing the initial basis

- “**Slack**” basis is simple choice $\mathbf{x}_0$
- Standard **crash** aims for feasible vertex $\mathbf{x}_F$
- “**Idiot**” crash aims for near-optimal point $\bar{\mathbf{x}}^*$
- Idiot crash exists as code in `clp`
 - What is it?
 - (Why) does it work?
 - How good is it?
- Google wanted to know!



Idiot crash: What was known?

- **Definition:** Forrest (2002)
 - Source code of `clp`
- **Dissemination:** Forrest (2014)
 - “I gave a bad talk on it years ago”
 - “You minimize $\mu \cdot \text{objective} + \text{sum of squared primal infeasibilities}$ ”
 - “This is done column by column... you just solve a quadratic to get new value”
 - “Periodically you reduce μ ”
- **Analysis:** Forrest (2014)
 - “For many problems you finish with a small sum of infeasibilities and an objective a bit higher than the optimal one”

Idiot crash: Sounds familiar?

minimize $f(\mathbf{x})$ subject to $\mathbf{r}(\mathbf{x}) = \mathbf{0}$

Quadratic penalty method

- Minimize
$$\phi(\mathbf{x}, \mu) = f(\mathbf{x}) + \frac{1}{2\mu} \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x})$$
- Decreasing sequence $\{\mu^k\}$
- $\mathbf{x}^k \rightarrow \mathbf{x}^*$ as $k \rightarrow \infty$
- Subproblems increasingly ill-conditioned as μ^k decreases

Beale (1985)

- Quadratic form minimization as LP crash
- Implemented in SCICONIC

Augmented Lagrangian method

- Minimize
$$\phi(\mathbf{x}, \mu) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{r}(\mathbf{x}) + \frac{1}{2\mu} \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x})$$
- Decreasing sequence $\{\mu^k\}$
- $\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \mu^k \mathbf{r}(\mathbf{x}^k)$
- $\mathbf{x}^k \rightarrow \mathbf{x}^*$ and $\boldsymbol{\lambda}^k \rightarrow \boldsymbol{\lambda}^*$ rapidly so ill-conditioning not an issue

Idiot algorithm

- Starts like augmented Lagrangian
- Finishes like quadratic penalty method

Idiot crash: Algorithm

$$\text{minimize } f = \mathbf{c}^T \mathbf{x} \quad \text{subject to } A\mathbf{x} = \mathbf{b} \quad \mathbf{x} \geq \mathbf{0}$$

The Idiot algorithm: $\mathbf{r}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$

Initialize $\mathbf{x}^0 \geq \mathbf{0}$, μ^1 , $\lambda^1 = \mathbf{0}$

For $k = 1, 2, 3, \dots K$

$$\mathbf{x}^k = \arg \min_{\mathbf{x} \geq \mathbf{0}} h(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \lambda^k{}^T \mathbf{r}(\mathbf{x}) + \frac{1}{2\mu^k} \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x})$$

Possibly update μ :

$$\mu^{k+1} = \mu^k / \omega, \text{ for some factor } \omega > 1$$

$$\lambda^{k+1} = \lambda^k$$

Else update λ :

$$\mu^{k+1} = \mu^k$$

$$\lambda^{k+1} = \mu^k \mathbf{r}(\mathbf{x}^k)$$

End

Idiot crash: Algorithm

- Solve subproblem

$$\min_{\mathbf{x} \geq \mathbf{0}} h(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^k{}^T \mathbf{r}(\mathbf{x}) + \frac{1}{2\mu^k} \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \quad \text{where} \quad \mathbf{r}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$$

approximately by repeated component-wise minimization

- Initially
 - Penalty parameter μ^1 ranges between 0.001 and 1000
 - Perform 20-30 “sample iterations”, minimizing component-wise twice
 - Possibly abandon Idiot if 10% primal infeasibility reduction not achieved
- Then, according to LP dimensions
 - Number of iterations K ranges between 30 and 200
 - μ is reduced (every 3 or 6 iterations) by $\omega = 0.333$ (typically)
- Final value of μ is around machine precision
- $\boldsymbol{\lambda}^k \rightarrow \mathbf{0}$ rapidly

Idiot crash: How effective is it?

Results: Speed-up of the c1p primal simplex solver when the Idiot crash is used and the percentage of solution time accounted for by the Idiot crash

Best			Worst		
Model ¹	Speed-up	Idiot (%)	Model ¹	Speed-up	Idiot (%)
Linf_520c	9.4	8.2	FOME12	1.1	0.1
STP3D	6.5	0.9	KEN-18	1.0	0.7
self	6.1	22.7	DFL001	1.0	0.1
STORM_1000	4.5	0.8	PDS-80	1.0	0.1
nug15	4.2	0.1	maros-r7	0.9	7.8
STORM-125	4.1	10.1	TRUSS	0.8	17.1

- Mean speed-up is 1.9; mean solution time accounted for by Idiot is 6%
- For only **some problems** does vanilla c1p use the Idiot crash and primal simplex

[1: Results drawn from experiments on 30 Mittelmann benchmarking problems]

Idiot crash: Effect on clp benchmark performance

Results: Performance of clp relative to cplex, gurobi and xpress

Mittelman (25/04/18)

Model	cplex	gurobi	xpress	clp
LINF_520C	495	574	255	35
NUG15	338	12	7	14
QAP12	26	1	1	5
QAP15	365	12	6	13
SELF	18	12	15	5

- For LINF_520C, clp is vastly faster
- For the three QAP linearizations, clp is very much faster than cplex
- For SELF, clp is significantly faster

Idiot crash: Can it solve LPs?

Results: Accuracy of final point after (up to) 200 Idiot iterations

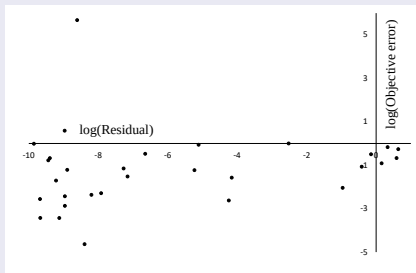
- Residual $\|A\mathbf{x} - \mathbf{b}\|_2$
- Objective error $\frac{|f - f^*|}{\max(1, |f^*|)}$

Best			Worst		
Model	Residual	Objective	Model	Residual	Objective
NUG15	2.1×10^{-10}	3.7×10^{-4}	DBIC1	3.8×10^{-1}	8.5×10^{-2}
MAROS-R7	4.0×10^{-9}	2.3×10^{-5}	STORM-125	1.4×10^0	1.2×10^{-1}
PDS-100	7.6×10^{-10}	3.7×10^{-4}	TRUSS	7.1×10^{-1}	3.2×10^{-1}
QAP15	2.1×10^{-10}	2.8×10^{-3}	MOD2	3.9×10^0	2.1×10^{-1}
LP22	1.1×10^{-9}	1.3×10^{-3}	PILOT87	2.1×10^0	6.8×10^{-1}
DFL001	1.1×10^{-9}	3.7×10^{-3}	WORLD	4.3×10^0	5.5×10^{-1}

- Idiot crash clearly solves some problems to acceptable tolerances
- Objective error measure using f^* is not an optimality test

Idiot crash: For what LPs does it work well?

Accuracy measure



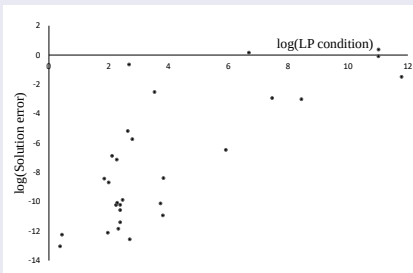
A convenient single quality measure for the point returned by the Idiot crash is

$$\text{qual}(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\| \times \frac{|f - f^*|}{\max(1, |f^*|)}$$

No problems with low value of $\text{qual}(\mathbf{x})$ have accurate optimal objective function value but large residual

Idiot crash: For what LPs does it work well?

Accuracy and condition



- Clear correlation between accuracy of final point and $\text{cond}(A)$
- Quadratic assignment problems are particularly well conditioned

Idiot crash: What can be proved?

$$\text{minimize } f = \mathbf{c}^T \mathbf{x} \quad \text{subject to } A\mathbf{x} = \mathbf{b} \quad \mathbf{x} \geq \mathbf{0}$$

The Idiot objective is bounded below for bounded LP problems

The Idiot objective $h^k(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \lambda^k r(\mathbf{x}) + \frac{1}{2\mu^k} \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x})$, where $\mathbf{r}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$, has positive semi-definite Hessian $A^T A$, but unboundedness of $h^k(\mathbf{x})$ implies unboundedness of the LP

The Idiot algorithm with exact solution of subproblems converges

Theorem: Suppose, that $\mathbf{x}^k$ is the exact global minimizer of $h^k(\mathbf{x})$ for each $k = 1, 2, \dots$ and that $\{\mu^k\} \rightarrow 0$ as $k \rightarrow \infty$. Then every limit point of the sequence $\{\mathbf{x}^k\}$ is a solution to the LP problem.

However: Subproblems are not (currently) solved exactly and ill-conditioning due to small μ mitigates against it

Idiot crash: What happens next?

- Final point is *not* a vertex (basic) solution
- clp performs
 - **Crossover** to get a basic solution
 - **Primal simplex** to get an optimal solution

Model	Speed-up	Idiot (%)	Residual	Objective
QAP15	4.0	0.1	2.1×10^{-10}	2.8×10^{-3}
NUG15	4.2	0.1	2.1×10^{-10}	3.7×10^{-4}
QAP12	2.5	0.6	3.6×10^{-10}	1.7×10^{-1}
STORM_1000	4.5	0.8	5.9×10^{-6}	5.9×10^{-2}
STP3D	6.5	0.9	7.0×10^{-5}	2.7×10^{-2}
PDS-100	2.5	5.4	7.6×10^{-10}	3.7×10^{-4}
LINF_520C	9.4	8.2	1.1×10^{-1}	9.1×10^{-3}

Idiot is a worthwhile crash, but relatively expensive to establish optimality!

Idiot crash: Bounding the optimal objective value

- **Know:** If the Idiot crash yields a feasible point $\bar{\mathbf{x}}^*$ then

$$f^* \leq \mathbf{c}^T \bar{\mathbf{x}}^* = \bar{f}^*$$

- **Consider:** dual problem

$$\text{maximize } f_D = \mathbf{b}^T \mathbf{y} \quad \text{subject to } A^T \mathbf{y} + \mathbf{s} = \mathbf{c} \quad \mathbf{s} \geq \mathbf{0}$$

- If the Idiot crash yields a feasible point $\bar{\mathbf{y}}^*$ then

$$\bar{f}_D^* = \mathbf{b}^T \bar{\mathbf{y}}^* \leq f^*$$

- Hence f^* lies in the interval $[\bar{f}_D^*, \bar{f}^*]$
- **Unfortunately:** Values of $\bar{f}_D^*$ don't (yet) have high accuracy

Idiot crash: Application to quadratic assignment problem linearizations

Quadratic assignment problem (QAP)

$$\min f(X) = \sum_{i,j,k,l} a_{ik} b_{jl} x_{ij} x_{kl} \quad \text{s.t. } X = [x_{ij}]_{n \times n} \in \Pi_n$$

This is a MIQP problem with n^2 binary variables and $2n$ constraints

QAP linearization (Adams and Johnson)

$$\begin{aligned} \min \quad & f(X) = \sum_{i,j,k,l} a_{ik} b_{jl} y_{ijkl} \\ \text{s.t.} \quad & \sum_i y_{ijkl} = x_{kl}, \quad j, k, l = 1, \dots, n; \quad \sum_j y_{ijkl} = x_{kl}, \quad i, k, l = 1, \dots, n \\ & y_{ijkl} \geq 0, \quad i, j, k, l = 1, \dots, n; \quad X = [x_{ij}]_{n \times n} \in \Pi_n \end{aligned}$$

This is a MILP problem with n^2 binary variables; n^4 continuous variables $y_{ijkl} = x_{ij} x_{kl}$ and $n^4 + 2n^3 + 2n$ constraints.

Idiot crash: Application to quadratic assignment problem linearizations

Results: Performance after (up to) 200 Idiot iterations

Model	Rows	Columns	Optimum	Residual	Objective	Error	Time
NUG05	210	225	50.00	9.4×10^{-9}	50.01	1.5×10^{-4}	0.04
NUG06	372	486	86.00	7.8×10^{-9}	86.01	1.2×10^{-4}	0.11
NUG07	602	931	148.00	7.9×10^{-9}	148.64	4.3×10^{-3}	0.25
NUG08	912	1613	203.50	7.0×10^{-9}	204.41	4.5×10^{-3}	0.47
NUG12	3192	8856	522.89	8.8×10^{-9}	523.86	1.8×10^{-3}	2.58
NUG15	6330	22275	1041.00	8.9×10^{-9}	1041.38	3.7×10^{-4}	5.13
NUG20	15240	72600	2182.00	7.5×10^{-9}	2183.03	4.7×10^{-4}	14.94
NUG30	52260	379350	4805.00	1.1×10^{-8}	4811.41	1.3×10^{-3}	82.28

- Solution of NUG30 intractable using simplex or IPM on the same machine
- Idiot crash consistently yields near-optimal solutions
- Useful within a branch-and-bound solver?

Exact Idiot

- Convergence of component-wise search can be prohibitively slow
- Solve

$$\min_{\mathbf{x} \geq \mathbf{0}} h(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \lambda^k{}^T \mathbf{r}(\mathbf{x}) + \frac{1}{2\mu^k} \mathbf{r}(\mathbf{x})^T \mathbf{r}(\mathbf{x})$$

directly using conjugate gradient based approach

- Preconditioning?

Parallel Idiot

- Approximate or exact Idiot dominated by cost of forming $A\mathbf{x}$
- “Obvious” parallelism is memory bound
- Problem-specific code?

Conclusions on the Idiot crash

- Has been presented in algorithmic form for the first time
- Generally beneficial for the primal revised simplex method
- Converges to an optimal solution when subproblems are solved exactly
- Consistently and quickly yields near-optimal solutions of QAP linearizations intractable with simplex or IMP

Slides: <http://www.maths.ed.ac.uk/hall/COA18>

Report: <http://www.maths.ed.ac.uk/hall/GaHa18>