Nikhil Sasi
2025-Dec

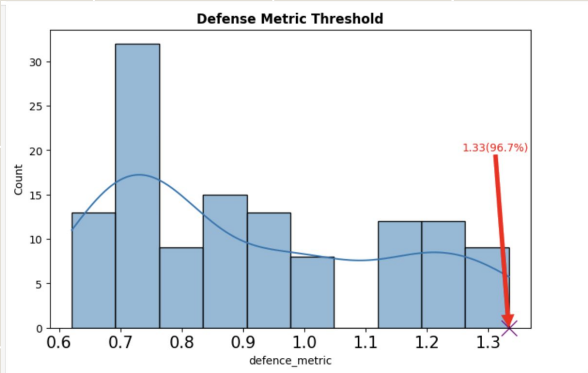
Linear Programming

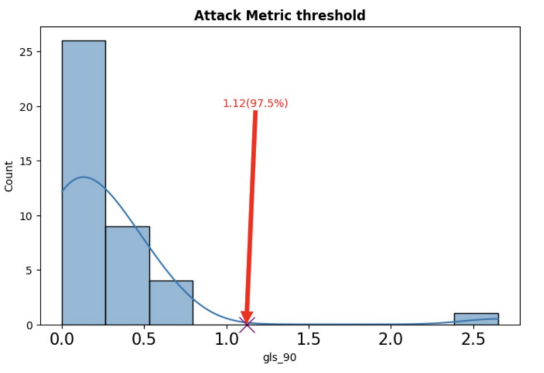
Application & algorithmic landscape

A example application and the types of LP approaches with their advantages and limitations

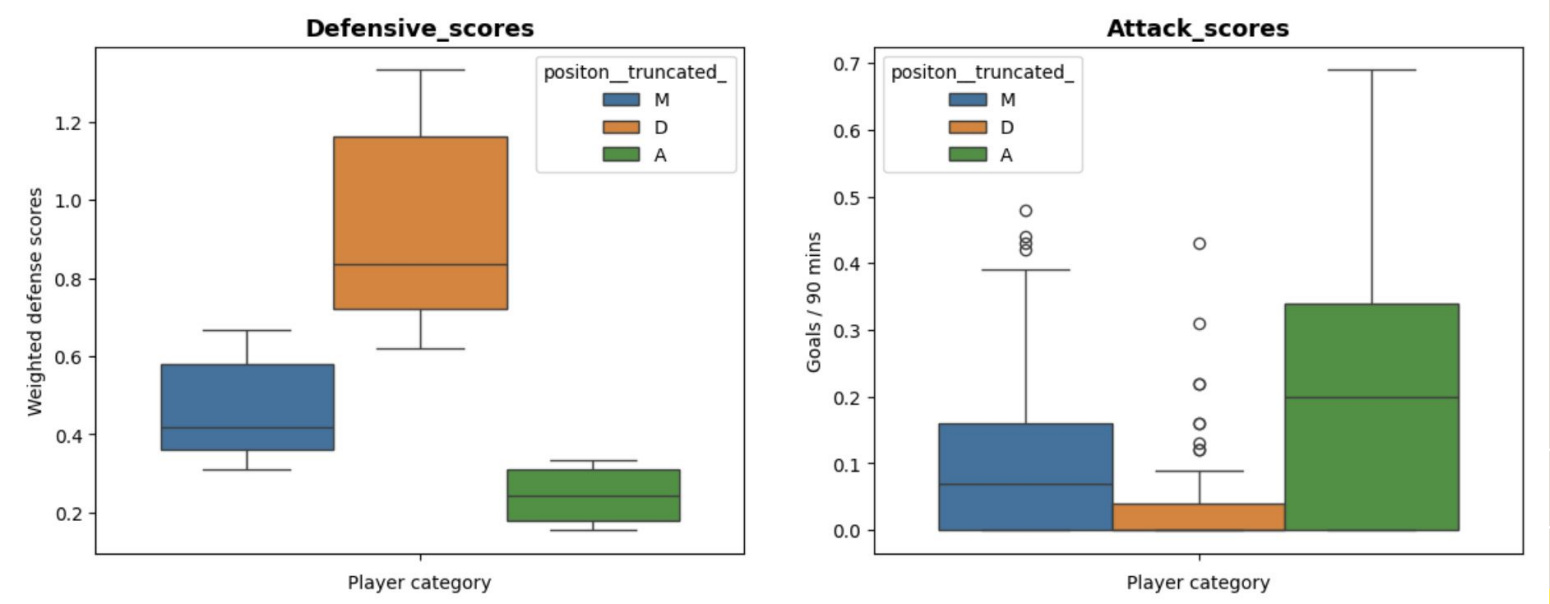
Moneyball - The application

- 296 Players considered from League (296 Decision variables)
  - Each decision variable value represents how likely player will be in team
- 3 constraints
  - Attack Power
  - Defense power
  - Size constraint
- Final team has 4 times top 2% of average defensive and average attacker metric as threshold (>98th percentile)



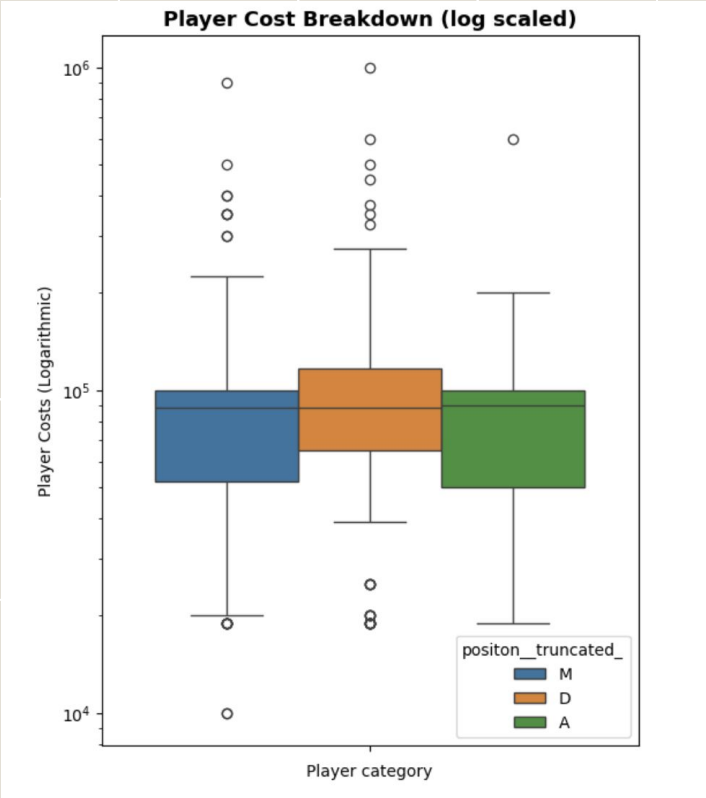


Performance Metric distributions



- Performance Metrics follow the role theme for Defense and not so strictly for Attack metrics.
- Defenders are great at defending.
- But attack power is more distributed across Attackers and Midfielders.

Objective distribution (Player cost)



- Player costs data is heavily skewed.
- But approximately evenly distributed across all 3 roles.
  - Since role can be a confounder this is a good sign.
  - Less likely one role may dominate due to pricing skewness
- We won't modify or crop the outliers since they represent real life extremes for objective coefficients.

Matrix representation

$$\text{MIN}_x \left(\begin{pmatrix} C_1 & C_2 & \dots & C_N \end{pmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_N \\ \beta_1 & \beta_2 & \dots & \beta_N \\ 1 & 1 & \dots & 1 \end{pmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \leq \begin{pmatrix} \text{Attack Power} \\ \text{Defence Power} \\ 10 \end{pmatrix}$$

Variables

- x_i : Team suitability value for player i ; $x \in [0, 1]$ (`DECISION VARIABLES` })
- C_i : Cost of player i
- α_i : Player i Attack power
- β_i : Player i Defence power

Limitations & Assumptions

Non Linearity not considered

Interaction terms $= \alpha_i.\beta_i$	Not Considered
Non linear terms $= \alpha_i^2$ or $1/\beta_i$	Not Considered

Independence

Presence of player 1 boosts Attack Metric of player 2	Not considered

Continuous decision variables only (instead of binary)

Simplex : $x \in [0,1]$ vs MILP / ILP : $x \in \{0,1\}$

The Moneyball Team

The final selected team, which exceeds 98% of random teams, consists of:

	player	squad	value	defence_metric	attack_metric
202	Maurice Nugent	Galway United	€20k	1.241379	0.310000
206	Michael Duffy	Derry City	€10k	0.580645	0.190000
36	Bridel Bosakani	Drogheda United	€72k	0.155172	2.650000
11	Al Amin Kazeem	St Patrick's	€25k	0.972973	0.430000
14	Alex Nolan	St Patrick's	€19k	0.486486	0.250000
227	Rayhaan Tulloch	Shelbourne FC	€19k	0.666667	0.310000
198	Mason Melia	St Patrick's	€25k	0.486486	0.370000
235	Rob Slevin	Galway United	€19k	1.241379	0.090000

Required : AP = 4.4848, **DP** = 5.3333

Achieved : AP = 4.6000, **DP** = 5.8312

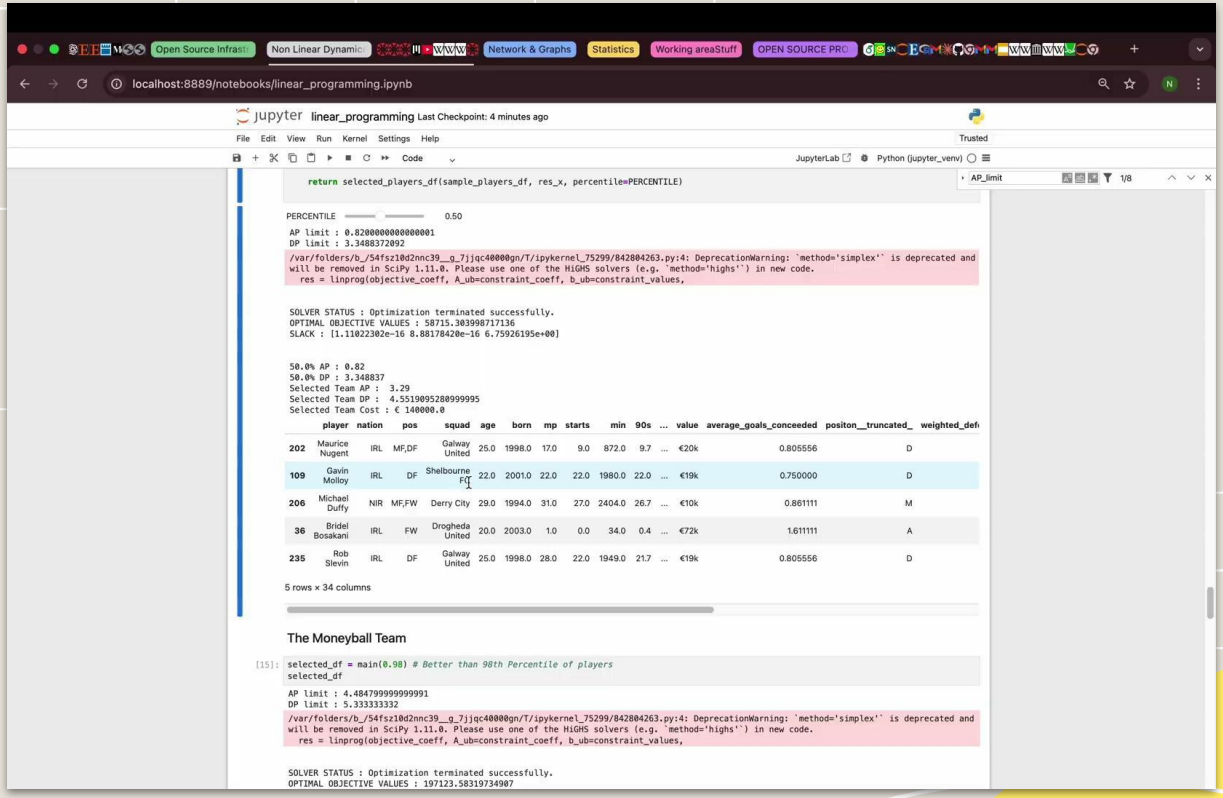
Objective = Team Acquisition Cost = $209,000

Slack values :

AP constraint ≈ 0 , **DP constraint** = 0, **Team size** = 2.625

Solver status : SOLVED, optimal found

LP in action (slider)



Linear Programming - standardisation, checks, preprocessing

How to standardise the inequality forms , identify common issues unsolvable in all LP problems and improve numerical stability for optimisation

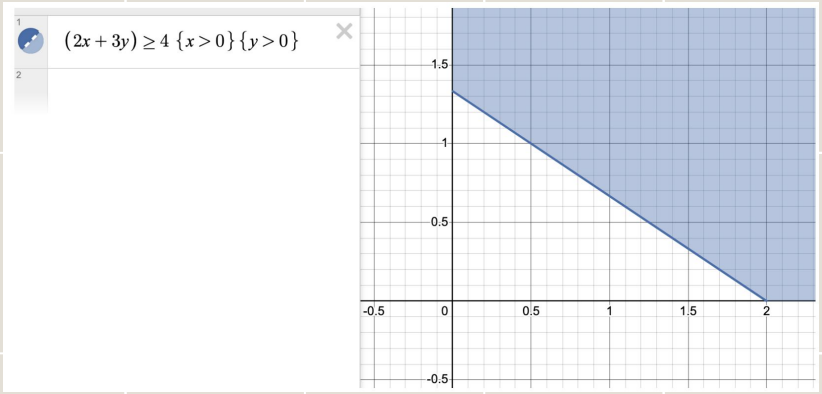
Standard form

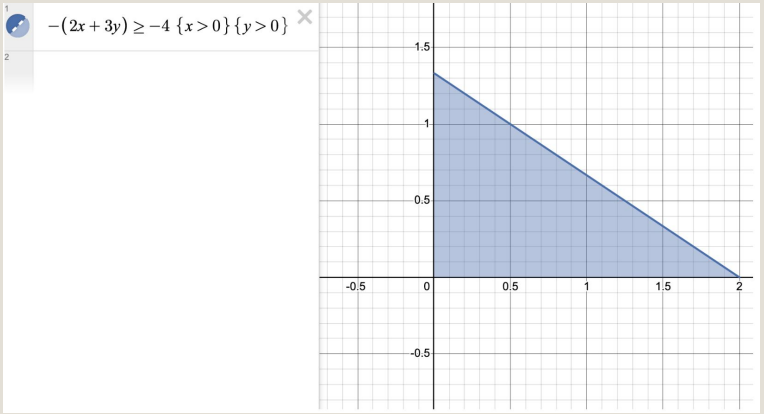
$$\min_x \; c^T x$$
$$\text{such that } A_{ub} x \leq b_{ub},$$

- Depends on solver requirements.
- Almost always require less than inequality (<=) in constraints.
- Objective can be minimized or maximized as a standard
- Example on left is from **Scipy.optimize**

- For most solvers and algorithms , we assume a standard form for the inequality.
- Linear programming problems can occur in multiple formats
  - Less than equal to inequalities (<=)
  - More than equal to inequalities (>=)
  - Equalities (=)
- The standard form generalises the approach of input so that optimisation can be approached in a common way for all problems.

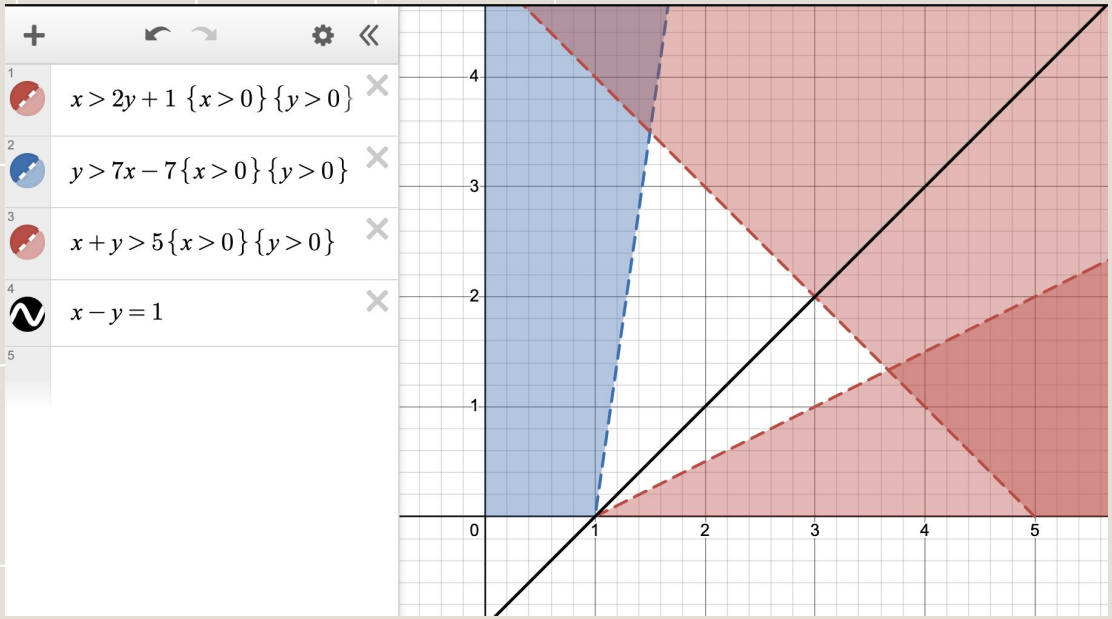
Standardisation - Conversion to standard form





- For greater than inequalities (>=) we multiply both sides with -1 to flip the Feasible region from unbounded to bounded , turns it into less than inequality (<=)
- For equalities (=), we split it into two inequalities
  - f(x) = K is split into
    - f(x) <= k
    - f(x) >= k

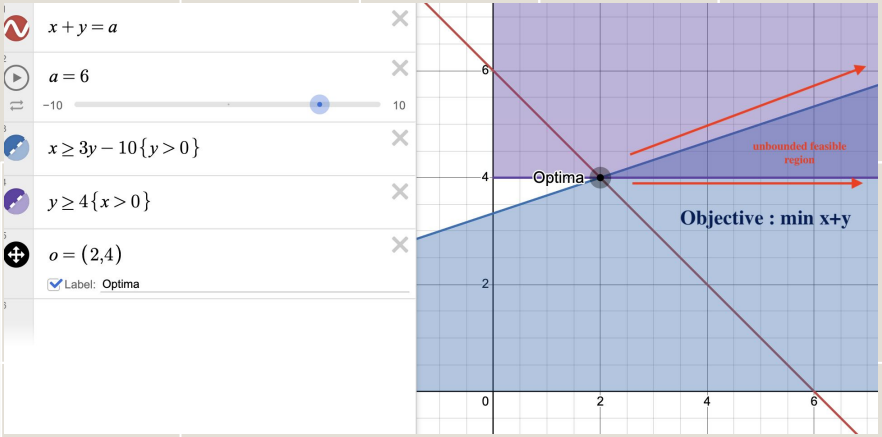
Feasible region - No Feasible region



There is no Feasible region which satisfies all constraints. Problems like these have no solutions. Constraints will have to be relaxed or the status is unsolvable

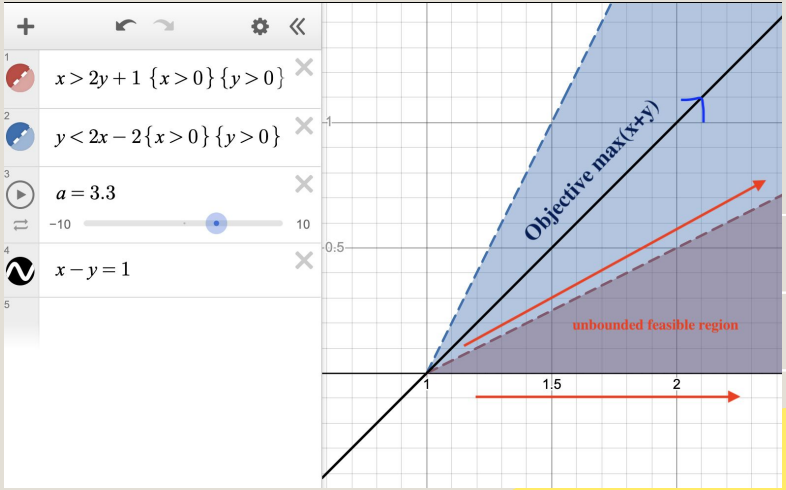
Feasible region - Unboundedness

Unbounded Feasible region finite optima



Happens when objective function vector direction is away from the unboundedness direction

Unbounded Feasible region non finite optima



Objective function points towards unbounded region. No finite optima value exists

Scaling (optional for Simplex but mandatory for IPM)

- Since LP computations entail a lot of arithmetic steps between coefficients of different scales for
  - Feasibility check
  - Pivoting
  - Newton factorisation for IPM
- Numerical stability is a very important part of the LP application
- Especially true for IPM methods which cannot function without Scaling in very large problems
  - Requires multiple matrix multiplications and this can cause the matrix to become unstable as many small multiplications happen
  - Round off errors and instability in Newton steps
- Scaling algorithms can
  - Recalibrate coefficients of objective, constraints and RHS on the same scale
  - Maintain equilibrium of matrices and rescale mid iteration
- Maintain numerical equilibrium throughout the LP operation and not just initial condition.

Linear Programming Approaches

A look on overall approach and research direction

Linear program algorithmic approaches

Walking Algorithms (Simplex Family)
The geometric approach

Interior Points Method Family
The gradient and Jacobian based approach

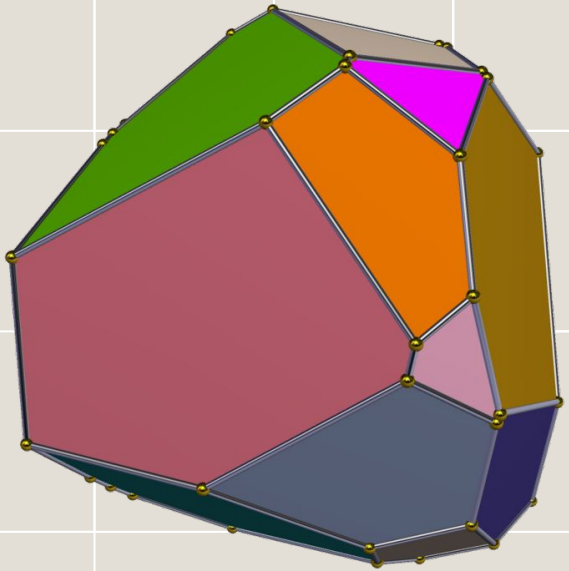
Plane cutting methods (derived from Ellipsoid approach)
Theoretical, doesn’t perform well in real life applications

Hybrid Approach
A combination of the above methods, usually a the IPM and walking algorithm fusion

Walking algorithmic approach (Simplex Family)

- Solving by geometry

Makes use of Geometry quirks



A 3d convex polytope

- Makes use of the idea that all LP feasible regions are a **complex polytope**
  - Extremely well studied structure
  - Every two points can be connected by an edge that will stay within the feasible region.
  - The BFS are all extreme points along their vector direction from origin.
  - The optima always lies in one of these extreme regions.
  - Most of the time there will be at least one edge at any vertex or BFS that will purely improve the objective function.
  - The optimization is Convex in nature.

Why still Relevant (from 1947 - present)

- **Real life Application Synergy :** Exponential time solvable in theory but works very well with real life applications
- **Computationally Simpler :** Avoids numerical requirements that IPM algorithms require and allows for Warm start. (Caveats included)
- **Good for Medium to Large LP :** Works well on Works very well for medium to large applications (many tens of thousands of decision variables and constraints)
- **Warm start application :** Really good for LP problems where LP has to be resolved after small perturbations. Restarting with last Optima as initial condition. (Memory)
- **Strong Heuristics support :** Heuristics or geometrical tricks allow us to boost its performance and avoid many of its possible problems like
    - Dual optimisation for great performance boosts in application
    - Algorithmic bypasses for Degeneracy
    - Dimensionality reduction using Facial reduction
    - Scaling

The Dual

Primal Problem

Maximize $Z = \mathbf{cx},$

subject to

$$\mathbf{Ax} \leq \mathbf{b}$$

and

$$\mathbf{x} \geq \mathbf{0}.$$

Dual Problem

Minimize $\mathbf{W} = \mathbf{yb},$

subject to

$$\mathbf{yA} \geq \mathbf{c}$$

and

$$\mathbf{y} \geq \mathbf{0}.$$

- The dual approach allows us to transform the Linear programming problem (originally called Primal) into its Dual.
- Duality is based on the **Lagrangian form** and is essentially a restructuring of the Linear programming into a different Linear program with same optima.
- Dual Simplex is a methodology that solves the dual problem if the duals constraints are easier
- In application duals are often easier to solve than the Primal when many constraints exists.

The Dual Approach

$$\mathcal{L}(x, \lambda) = f(x) + \lambda g(x)$$

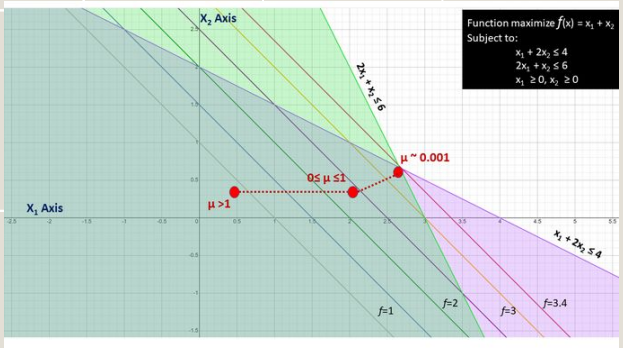
- The basis of the dual is the existence of **Lagrangian Multipliers**
- The above equation is the Lagrangian form for the optimisation problem
  - It combines the constraint and objective function in a single expression where the constraints offset the objective function based on constraints
  - The lagrange multiplier is the term **λ** and shows us how that particular constraint effect's the objective function.
  - It also opens up this new area of **Shadow prices** which tells us which decision variables have the most effect on the Objective function or the least.

Dual as a concept : https://nickarnosti.com/blog/lp_duality/

IPM family of approaches

Using Gradients, Lagrangians and matrix operations on a massive scale

Interior Point Methods - general overview



- Interior point methods take a central path from within the Feasible region.
- Each constraint is turned into a penalisation on the objective function. Like a Lagrangian expression.
- The gradient of the Lagrangian is taken and steps made on these gradients towards the optima.
- There is a barrier term that controls the magnitude of penalisation and this is reduced gradually so that the steps can converge on the optima near the constraint barrier.

Relevancy and strengths

- **Great efficiency with sparse data sets :** IPMs perform exceptionally well with sparse datasets where most of the constraint coefficients are 0. Scales really well.

- **Greater efficiency for very large problems :** for most very large industrial data sets the constraint matrix is sparse, so in application they outperform simplex for large to very large LP problems. Problems with millions of constraints and decision variables.

- **Immune to Degeneracy and Stalling :** IPMs circumvent the geometry of optimisation by numerical penalisation for approaching constraint edges. They are disassociated with the geometry of the Feasible region to a degree

- **Numerous IPM approaches and Research direction :** The IPM family takes Newton steps and there are lots of approaches and research areas possible to optimise step approach.

Hybrid Approaches

- Hybrid approaches are algorithms that often combine strengths of the two algorithmic approaches to maximise performance
- IPMs ability to traverse more efficiently in very large constraint problems and Simplex's ease in computation and geometric efficiency
- For example an approach where IPM crosses large areas in the feasible space and then switches to a Simplex adjacent algorithm near to the optima
- Implemented for more non sparse high dimensional LP problems

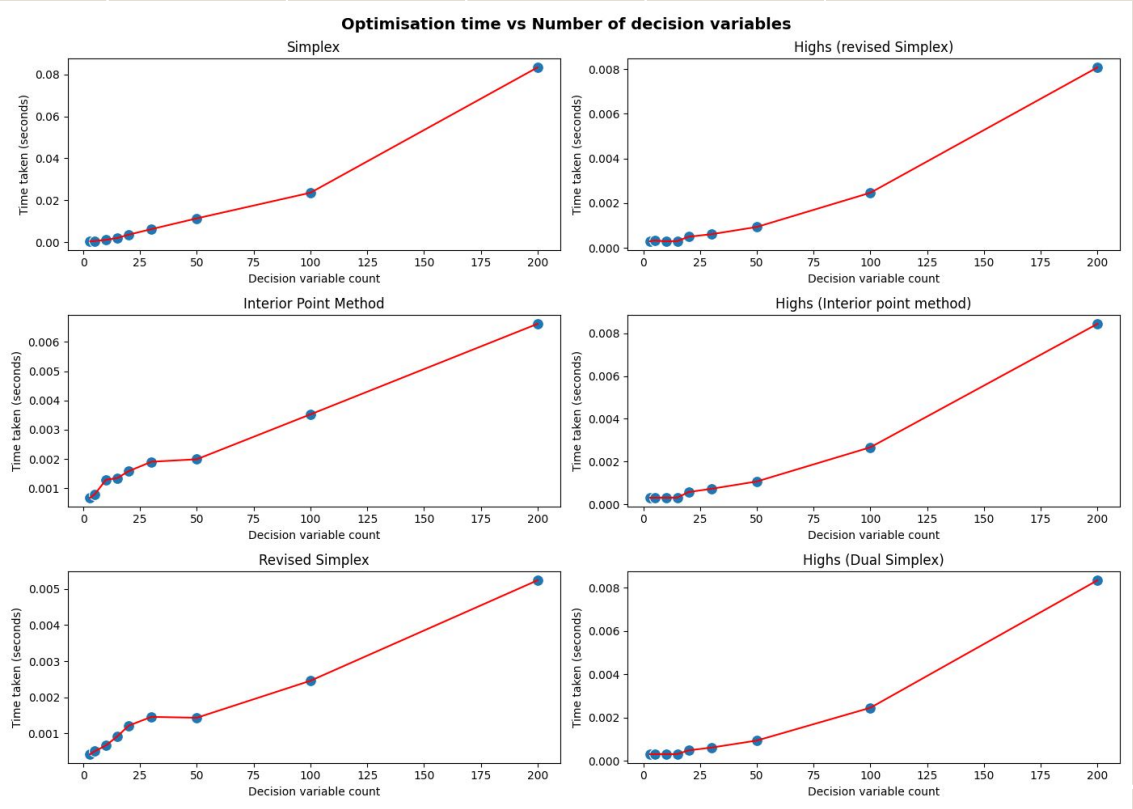
HiGHS (best Open source overall algorithm form University of Edinburgh)

- **Best overall open source :** HiGHS is the current overall best performing open source solver in the market

- **Preprocessing :** It has implementations for various preprocessing techniques like Scaling and Degeneracy handling built in.

- **Efficient IPM and dual revised simplex variants :** It uses the algorithmically efficient forms of simplex and IPM for greater solvability and efficiency

Time Complexity



An analysis of performance time averages on the Moneyball data

- Revised simplex performs best overall
- Old Simplex has the worst performance
- HiGHS performs overall well but slightly slower than plain IPM and Simplex
  - Due to large overhead of presolver and postsolver operations
  - Performs much better on significantly larger problems

HiGHS compared to other commercial solvers

Scaled shifted (by 10 sec) geometric mean of runtimes

unscaled	1053	38.5	288	687	2296	3616	255
scaled	27.4	1	7.49	17.9	59.7	94.1	6.63
solved	40	65	52	51	33	32	52

probs	CLP	COPT	MOSEK	HiGHS	GLOP	SPLX	XOPT

More detailed look : https://plato.asu.edu/ftp/lpopt.html