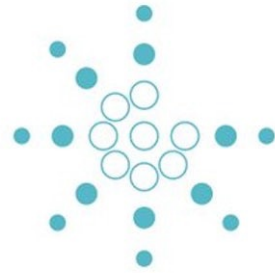


Multi-Objective Linear Programming to find the Best Pokémon Team

Dr Geraint Palmer
@GeraintPalmer



Gwerddon



Coleg
Cymraeg
Cenedlaethol

Rhaglennu llinol amlamcan i ganfod y tîm Pokémon gorau

Geraint Ian Palmer

Yr Adran Mathemateg, Prifysgol Caerdydd

Cyflwynwyd: 14 Rhagfyr 2021; Derbyniwyd: 21 Mawrth 2022

Crynodeb: Mae'r erthygl hon yn rhoi enghraifft o gymhwyso technegau ymchwil weithredol trwy optimeiddio timau Pokémon. Fformiwleiddir y broblem fel problem rhaglennu llinol amlamcan, a chanfyddir y ffyrnt Pareto trwy ddefnyddio'r feddalwedd PuLP yn Python. Cymherir datrysiadau optimaidd Pareto trwy efelychiad Monte Carlo o frwydrau Pokémon.

Allweddeiriau: rhaglennu llinol amlamcan, efelychu Monte Carlo, y ffyrnt Pareto, Pokémon, ymchwil weithredol.



DOT MATRIX WITH STEREO SOUND

MAGNETON

:L61



CHARIZARD

:L57



Enemy MAGNETON

used THUNDER WAVE!

Nintendo GAME BOY™





- 739 Pokémon
- 841 Moves (≈ 68 per Pokémon)
- Teams of 6
- 4 Moves per Pokémon



7.5×10^{18}
Grains of sand
on Earth



$${}^{739}C_6 \times {}^{68}C_4 \approx$$

1.8×10^{20}
Teams in Pokémon
Sword and Shield



1×10^{22}
Stars in the
observable universe



Linear Programming

I make £50 profit on a gallon of paint A, and £60 profit on a gallon paint B.

There are two ingredients: X and Y:

- A gallon of paint A requires 4 pints of X and 5 pints of Y.
- A gallon of paint B requires 6 pints of X and 4 pints of Y.

Only 24 pints of X and 20 pints of Y are available per day.

How much paint should I produce daily to maximise profit?

$$\textit{Maximise } 50A + 60B$$

Subject to:

$$4A + 6B \leq 24$$

$$5A + 4B \leq 20$$

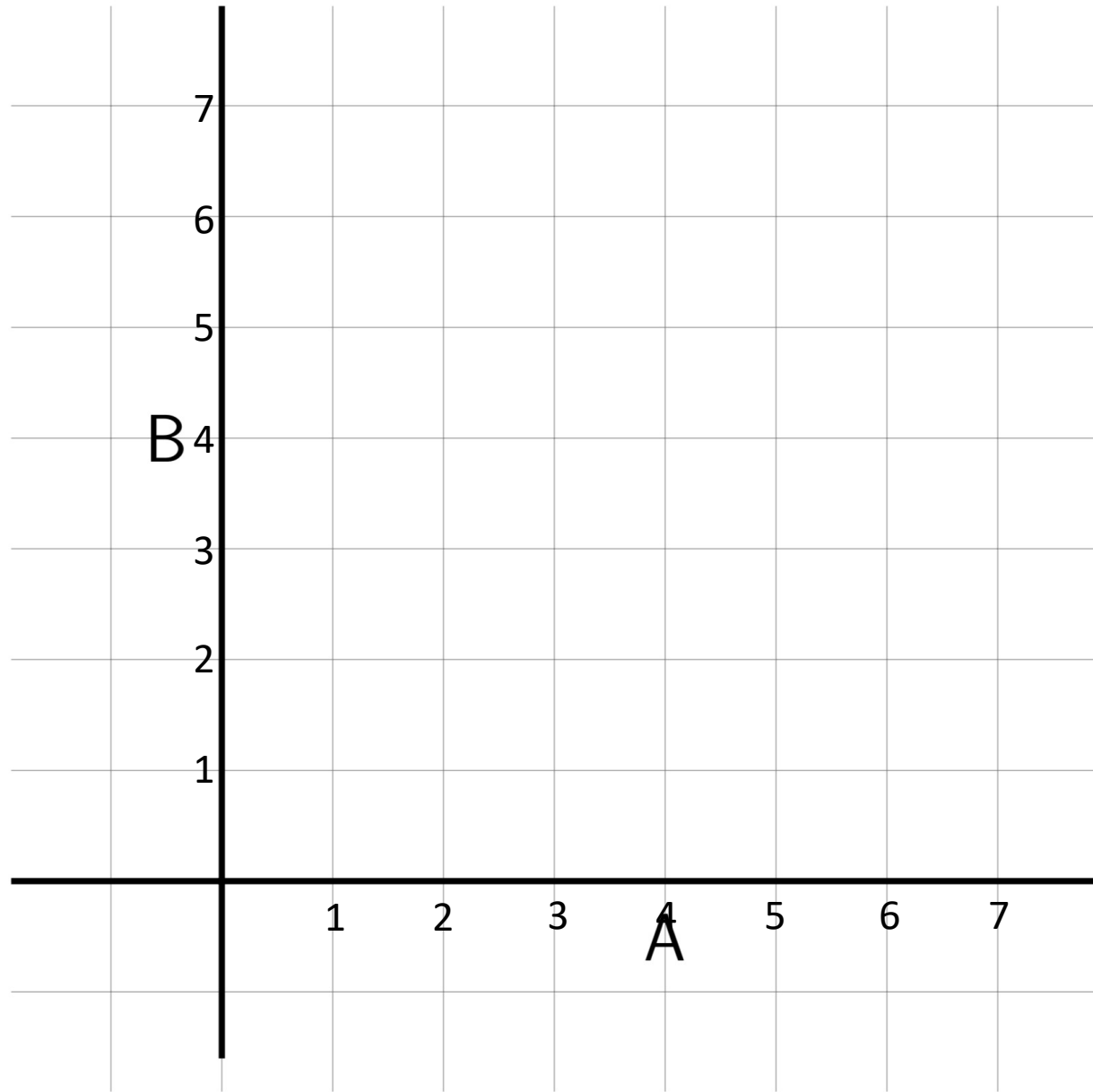


Maximise $50A + 60B$

Subject to:

$$4A + 6B \leq 24$$

$$5A + 4B \leq 20$$

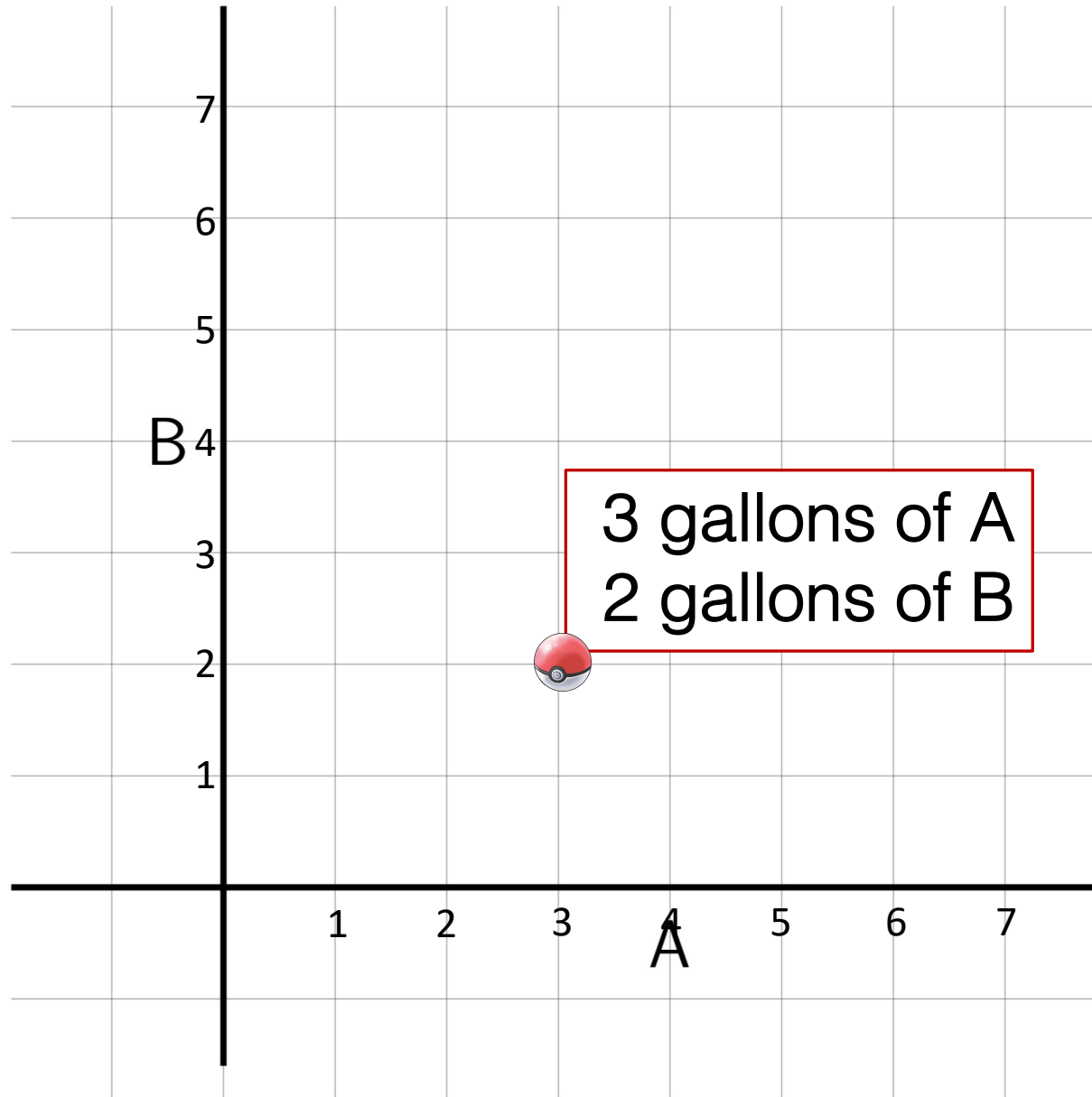


Maximise $50A + 60B$

Subject to:

$$4A + 6B \leq 24$$

$$5A + 4B \leq 20$$

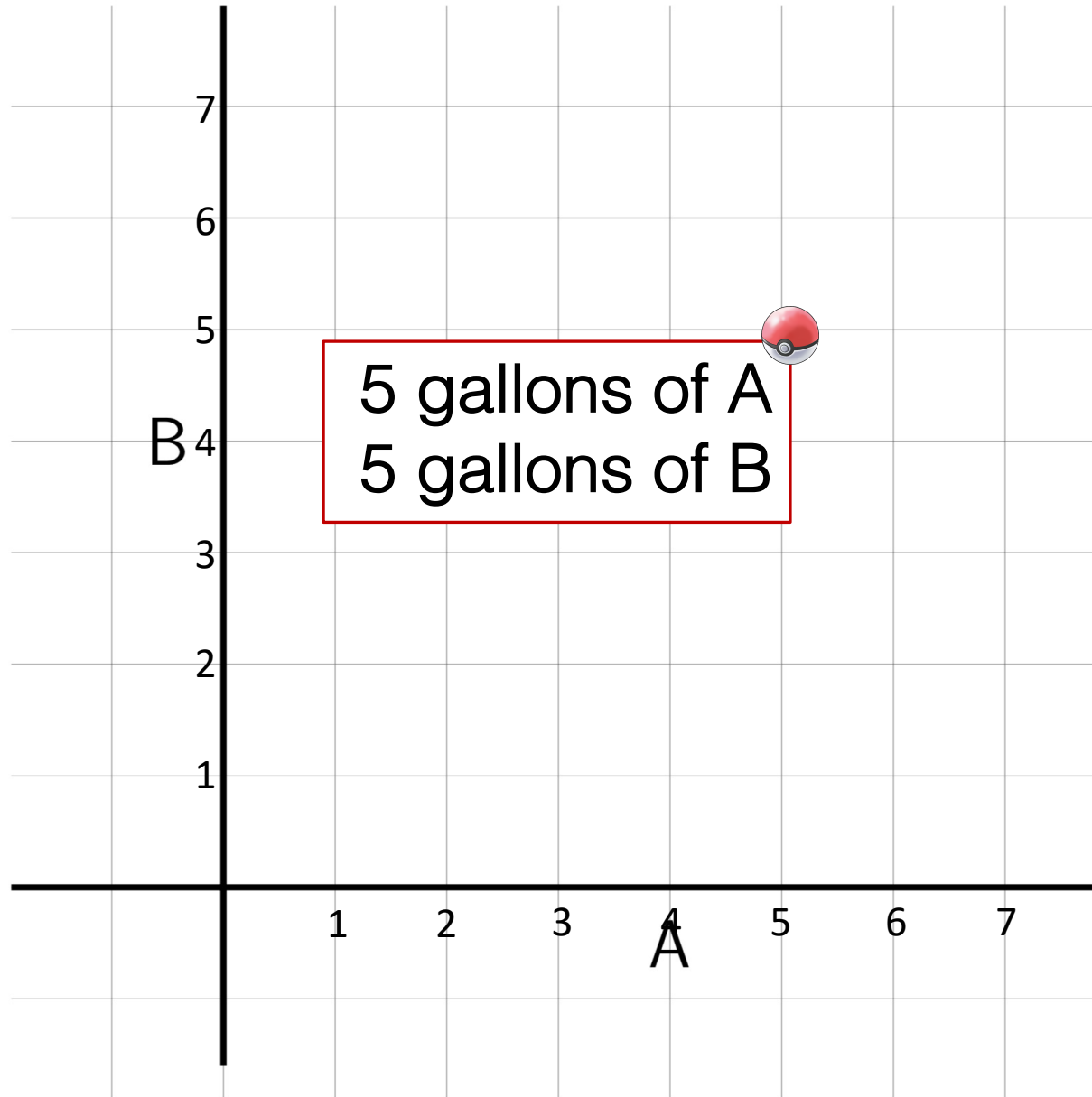


Maximise $50A + 60B$

Subject to:

$$4A + 6B \leq 24$$

$$5A + 4B \leq 20$$

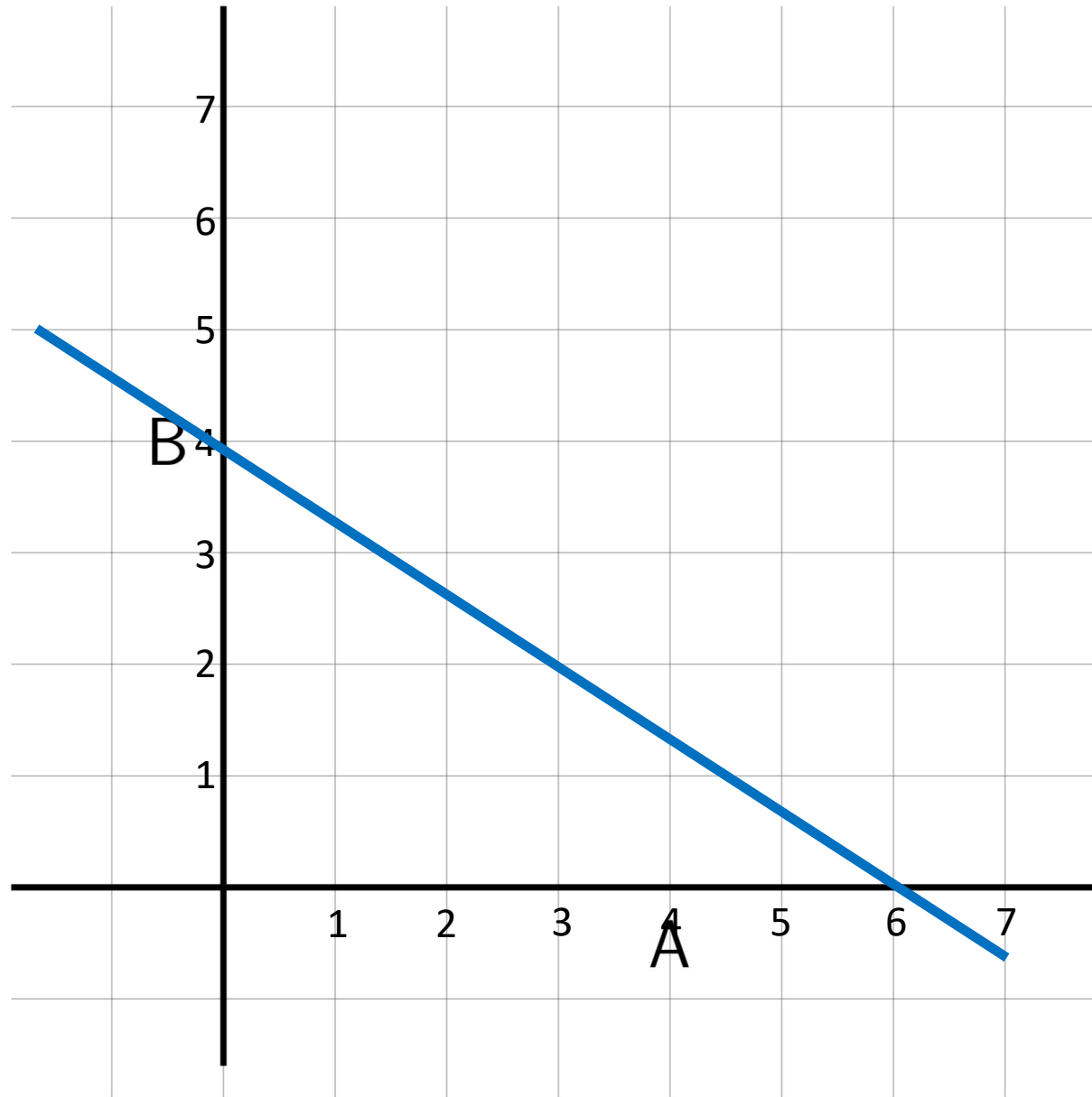


Maximise $50A + 60B$

Subject to:

$$4A + 6B \leq 24$$

$$5A + 4B \leq 20$$

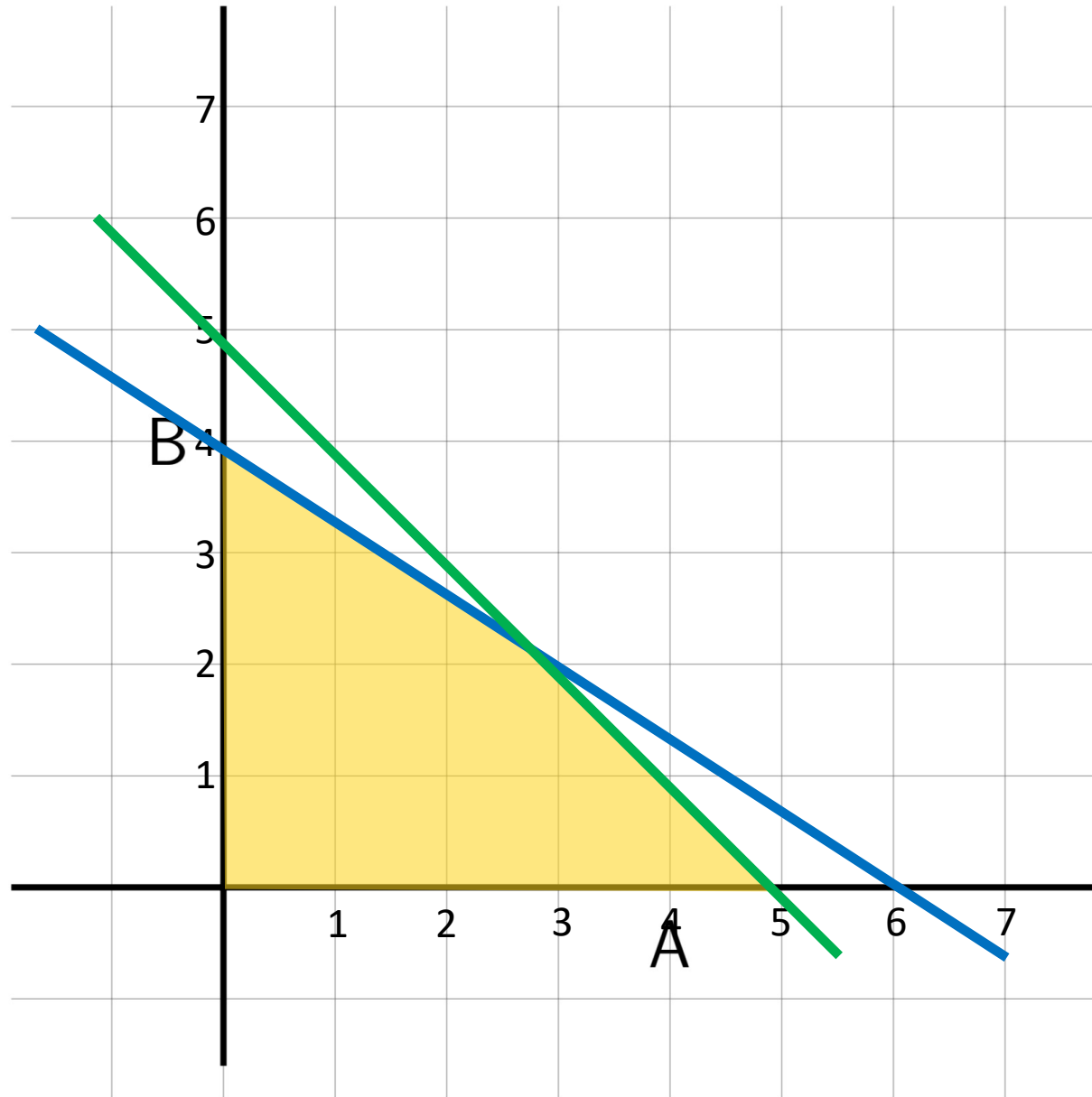


Maximise $50A + 60B$

Subject to:

$$4A + 6B \leq 24$$

$$5A + 4B \leq 20$$

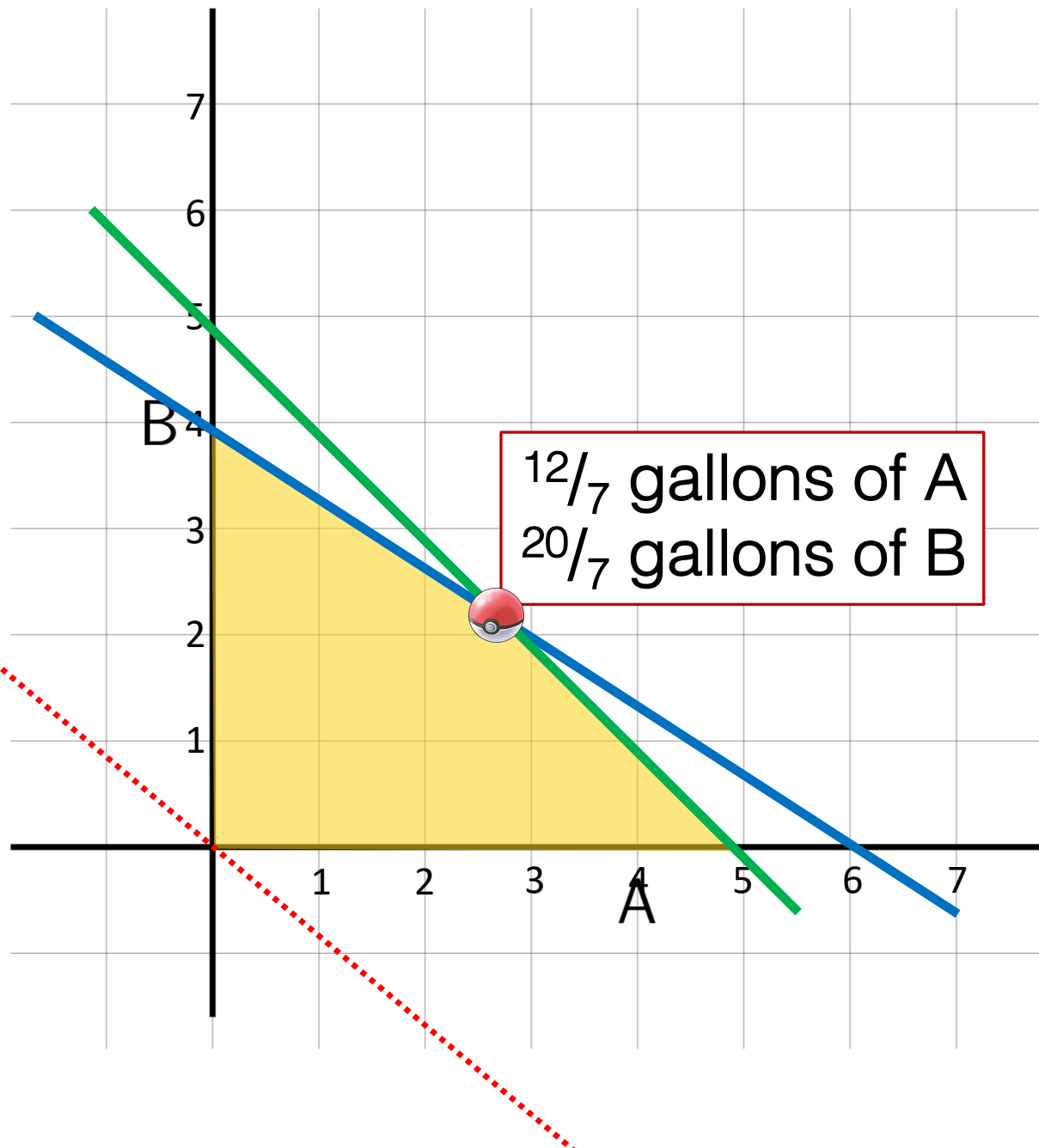


Maximise $50A + 60B$

Subject to:

$$4A + 6B \leq 24$$

$$5A + 4B \leq 20$$



```
import pulp

# Create problem
prob = pulp.LpProblem("Paint", pulp.LpMaximize)

# Create decision variables
paint = pulp.LpVariable.dicts("x", range(2))

# Objective function
objective_function = (5 * paint[0]) + (6 * paint[1])
prob += objective_function

# Constraints
prob += (4 * paint[0]) + (6 * paint[1]) <= 24
prob += (5 * paint[0]) + (4 * paint[1]) <= 20

# Solve
prob.solve()
print(paint[0].value(), paint[1].value())
```

1.7142857 2.8571429



Linear Programming



Integer Programming



Multi-Objective Programming



Stochastic Programming



Binary & Integer Programming

```
X = pulp.LpVariable.dicts("X", range(5), cat="Integer")
```

```
X = pulp.LpVariable.dicts("X", range(5), cat="Binary")
```



Muilt-Objective: Weighted Sum Method

f_1 = objective function 1

f_2 = objective function 2

w = weight between 0 and 1

$$\text{Maximise: } wf_1 + (1 - w)f_2$$



Blastoise



Water

HP	79
Attack	83
Special Attack	85
Defense	100
Special Defense	105
Speed	78

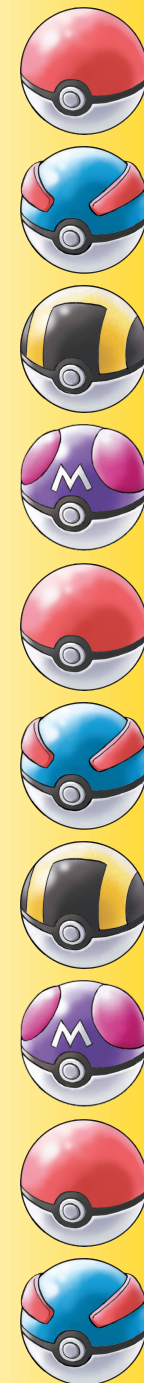
Special attacker

Hydro Pump Pwr: 110
(Water, special) Acc: 80%

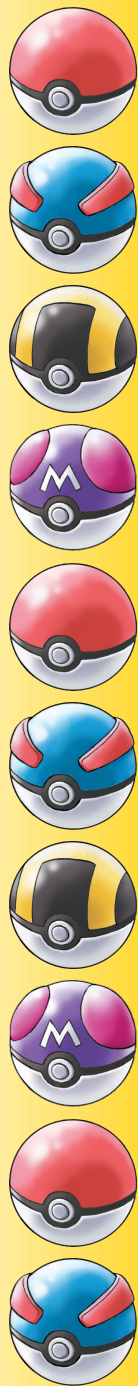
Bite Pwr: 60
(Dark, physical) Acc: 100%

Ice Beam Pwr: 90
(Ice, special) Acc: 100%

Surf Pwr: 90
(Water, special) Acc: 100%



		Move																		
		Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	Dark	Steel	Fairy	
Pokémon	Normal	1	1	1	1	1	1	2	1	1	1	1	1	1	0	1	1	1	1	
	Fire	1	½	2	1	½	½	1	1	2	1	1	½	2	1	1	1	½	½	
	Water	1	½	½	2	2	½	1	1	1	1	1	1	1	1	1	1	½	1	
	Electric	1	1	1	½	1	1	1	1	2	½	1	1	1	1	1	1	½	1	
	Grass	1	2	½	½	½	2	1	2	½	2	1	2	1	1	1	1	1	1	
	Ice	1	2	1	1	1	½	2	1	1	1	1	1	2	1	1	1	2	1	
	Fighting	1	1	1	1	1	1	1	1	1	2	2	½	½	1	1	½	1	2	
	Poison	1	1	1	1	½	1	½	½	2	1	2	½	1	1	1	1	1	½	
	Ground	1	1	2	0	2	2	1	½	1	1	1	1	½	1	1	1	1	1	
	Flying	1	1	1	2	½	2	½	1	0	1	1	½	2	1	1	1	1	1	
	Psychic	1	1	1	1	1	1	½	1	1	1	½	2	1	2	1	2	1	1	
	Bug	1	2	1	1	½	1	½	1	½	2	1	1	2	1	1	1	1	1	
	Rock	½	½	2	1	2	1	2	½	2	½	1	1	1	1	1	1	2	1	
	Ghost	0	1	1	1	1	1	0	½	1	1	1	½	1	2	1	2	1	1	
	Dragon	1	½	½	½	½	2	1	1	1	1	1	1	1	1	2	1	1	2	
	Dark	1	1	1	1	1	1	2	1	1	1	0	2	1	½	1	½	1	2	
	Steel	½	2	1	1	½	½	2	0	2	½	½	½	½	1	½	1	½	½	
Fairy	1	1	1	1	1	1	½	2	1	1	1	½	1	1	0	½	2	1		



Objective:

- 6 Pokémon with the highest stats;
- each with 4 moves maximising expected damage.

Such that:

- Can resist all types;
- Strong against all types;
- There is at least 2 special attackers;
- There is at least 2 physical attackers;
- Special attackers only know special moves;
- Physical attackers only know physical moves.



Decision Variables

$$X_p = \begin{cases} 1 & \text{if Pokémon } p \text{ is selected,} \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in \mathcal{P}$$

$$Y_{pm} = \begin{cases} 1 & \text{if Pokémon } p \text{ knows move } m, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in \mathcal{P}, m \in \mathcal{M}$$



Objective Function

- Maximises total stats;

$$f_1 = \sum_{p \in \mathcal{P}} T_p X_p$$

- Maximises possible expected damage;

$$f_2 = \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \left(1 + \frac{1}{2} \theta_{pm} \right) D_m A_m Y_{pm}$$

Maximise: $w f_1 + (1 - w) f_2$



Constraints

- Find a team of 6 Pokémon;

$$\sum_{p \in \mathcal{P}} X_p = 6$$

- With 4 moves each;

$$\sum_{m \in \mathcal{M}} Y_{pm} = 4X_p \quad \forall p \in \mathcal{P}$$



Constraints

- Pokémon only have moves they are allowed to learn;

$$Y_{pm} \leq L_{pm} \quad \forall p \in \mathcal{P}, \quad \forall m \in \mathcal{M}$$



Constraints

- There is at least 2 special attackers;
- There is at least 2 physical attackers;

$$\sum_{p \in \mathcal{P}} Z_p X_p \geq 2$$

$$\sum_{p \in \mathcal{P}} \Phi_p X_p \geq 2$$



Constraints

- Special attackers only know special moves;
- Physical attackers only know physical moves;

$$\sum_{m \in \mathcal{M}} (1 - Z_p) S_m Y_{pm} = 0 \quad \forall p \in \mathcal{P}$$

$$\sum_{m \in \mathcal{M}} (1 - \Phi_p) F_m Y_{pm} = 0 \quad \forall p \in \mathcal{P}$$



Constraints

- For each type there is at least one Pokémon that resists moves of that type;

$$\sum_{p \in \mathcal{P}} R_{pt} X_p \geq 1 \quad \forall t \in \mathcal{T}$$



Constraints

- For each type there is at least one Pokémon not weak to that type, that knows at least one move that is strong against that type;

$$\sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} (1 - W_{pt}) C_{mt} Y_{mp} \quad \forall t \in \mathcal{T}$$



$$f_1 = \sum_{p \in \mathcal{P}} T_p X_p$$

$$f_2 = \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \left(1 + \frac{1}{2} \theta_{pm}\right) D_m A_m Y_{pm}$$

maximise $w \frac{f_1}{\Psi_1} + (1 - w) \frac{f_2}{\Psi_2}$

subject to

$$\sum_{p \in \mathcal{P}} X_p = 6$$

$$\sum_{m \in \mathcal{M}} Y_{pm} = 4X_p \quad \forall p \in \mathcal{P}$$

$$Y_{pm} \leq L_{pm} \quad \forall p \in \mathcal{P}, \forall m \in \mathcal{M}$$

$$\sum_{p \in \mathcal{P}} Z_p X_p \geq 2$$

$$\sum_{p \in \mathcal{P}} \Phi_p X_p \geq 2$$

$$\sum_{m \in \mathcal{M}} (1 - Z_p) S_m Y_{pm} = 0 \quad \forall p \in \mathcal{P}$$

$$\sum_{m \in \mathcal{M}} (1 - \Phi_p) F_m Y_{pm} = 0 \quad \forall p \in \mathcal{P}$$

$$\sum_{p \in \mathcal{P}} R_{pt} X_p \geq 1 \quad \forall t \in \mathcal{T}$$

$$\sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} (1 - W_{pt}) C_{mt} Y_{pm} \geq 1 \quad \forall t \in \mathcal{T}$$



PuLP

```
prob = pulp.LpProblem("BestPokémonTeam", pulp.LpMaximize)

# Create decision variables
X = pulp.LpVariable.dicts("X", pdata.index, cat="Binary")
Y = pulp.LpVariable.dicts("Y", indexes, cat="Binary")

# Create objective functions
w = 0.5
f1 = (sum(X[p] * pdata.loc[p, 'TotalStats'] for p in pdata.index))
f2 = sum((1 + (0.5 * stab[p][a])) * mdata.loc[a, 'Power'] * mdata.loc[a, 'Accuracy'] * Y[(p, a)] for p in pdata.index for a in pmoves[p])
objective_function = (w * (f1 / maxf1)) + ((1 - w) * (f2 / maxf2))
prob += objective_function

# Create constraints
prob += sum(X[p] for p in pdata.index) == 6
prob += sum(Y[(p, a)] for p in pdata.index for a in pmoves[p]) == 24
prob += sum(X[p] * pdata.loc[p]['Special Attacker'] for p in pdata.index) >= 2
prob += sum(X[p] * pdata.loc[p]['Physical Attacker'] for p in pdata.index) >= 2
for p in pdata.index:
    prob += sum(Y[(p, a)] for a in pmoves[p]) == (4 * X[p])
    prob += sum(Y[(p, a)] * (1 - pdata.loc[p, 'Special Attacker']) * mdata.loc[a, 'Special'] for a in pmoves[p]) <= Gamma
    prob += sum(Y[(p, a)] * (1 - pdata.loc[p, 'Physical Attacker']) * mdata.loc[a, 'Physical'] for a in pmoves[p]) <= Gamma
for t in types:
    prob += sum(Y[(p, a)] * (1 - weaknesses[p][t]) * strong_against[a][t] for p in pdata.index for a in pmoves[p]) >= 1
    prob += sum(X[p] * resistances[p][t] for p in pdata.index) >= 1

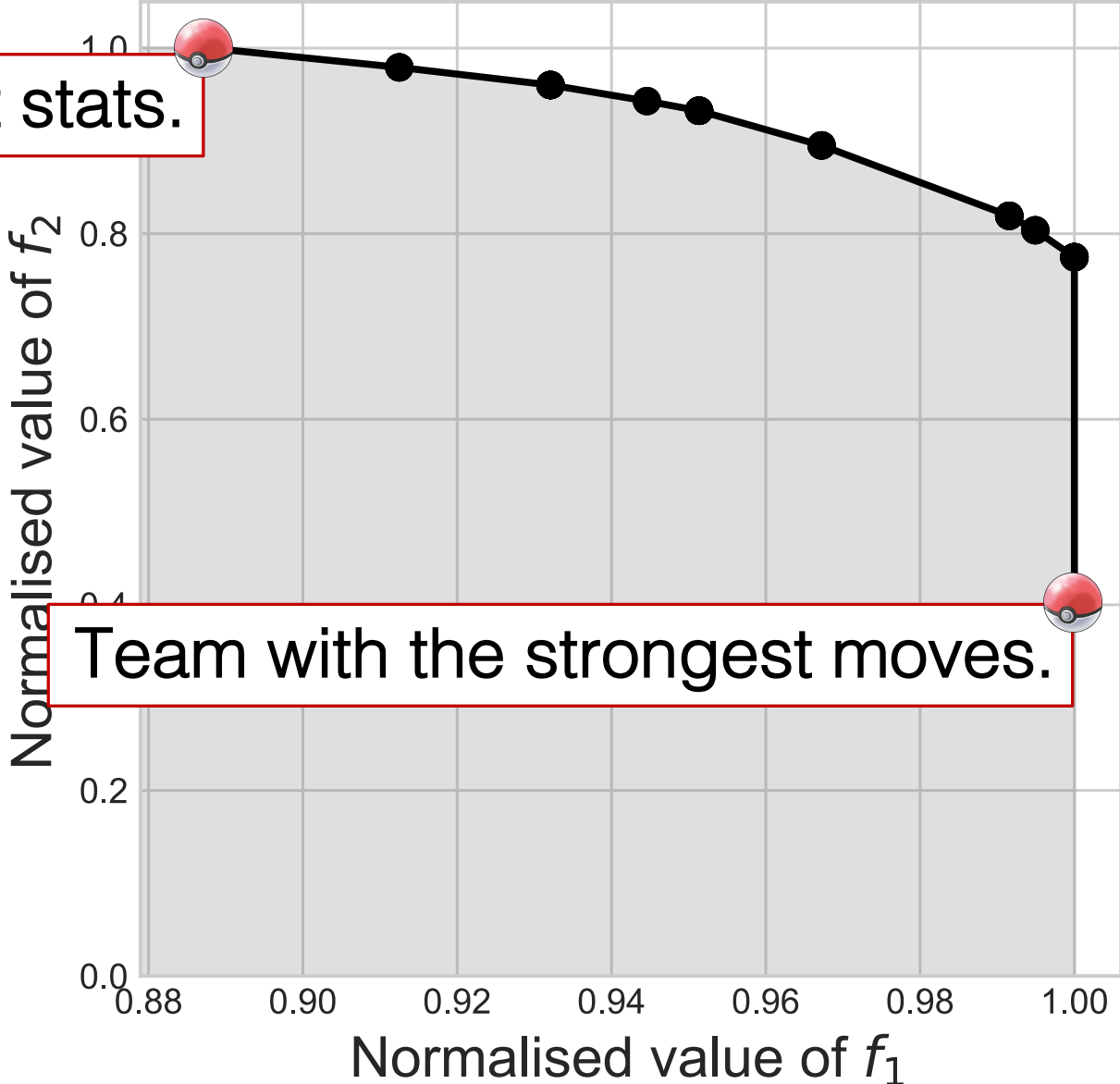
# Solve
prob.solve()
```



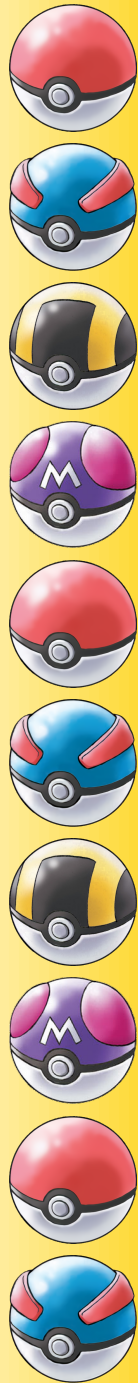
Pareto Front

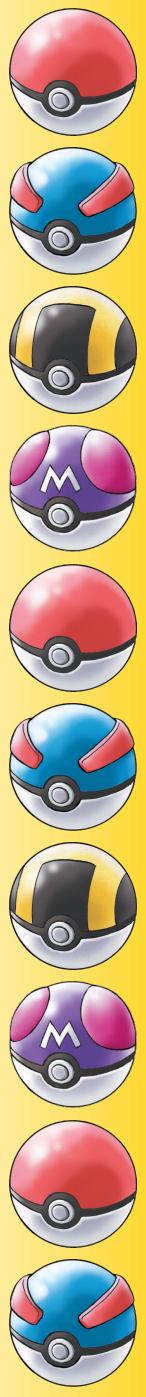
Objective Space

Team with the highest stats.



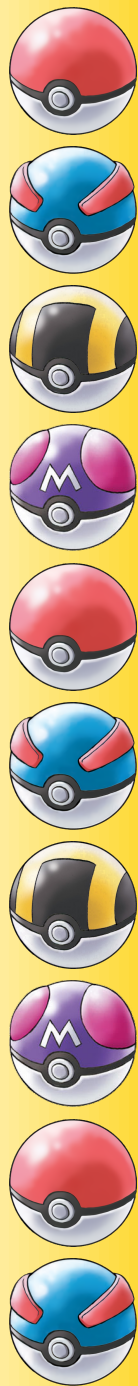
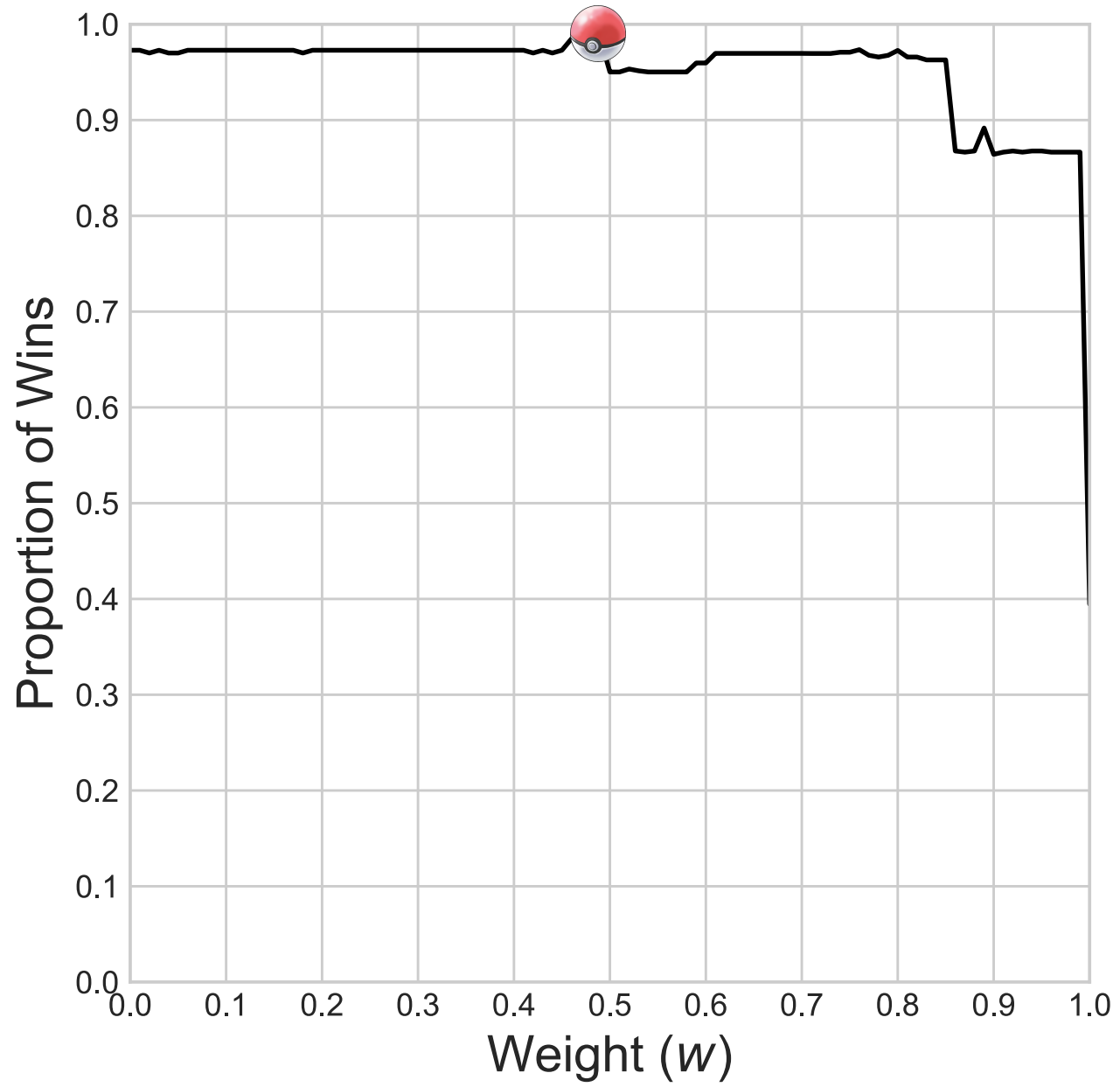
Team with the strongest moves.

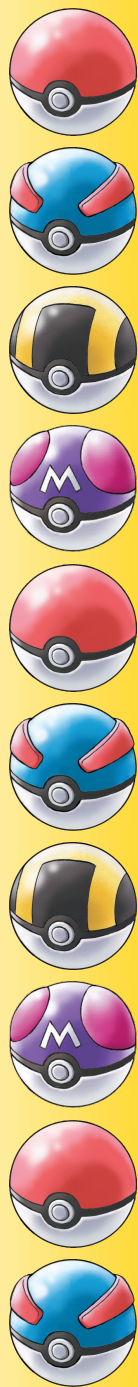




(simulated)
Battle!







Venusaur



Frenzy Plant
Leaf Storm
Petal Dance
Solar Beam

Blastoise



Hydro Canon
Hyper Beam
Surf
Water Spout

Scrafty



Close Combat
Focus Punch
Foul Play
High Jump Kick

Braviary



Brave Bird
Giga Impact
Sky Attack
Thrash

Aggron



Earthquake
Head Smash
Outrage
Steel Roller

Bewear



Close Combat
Trash
Giga Impact
Double-Edge



Constraints

- No Pokémon has more than one move of the same type.

$$\sum_{m \in \mathcal{M}} G_{mt} Y_{pm} \leq 1 \quad \forall p \in \mathcal{P} \quad \forall t \in \mathcal{T}$$



Blaziken



Brave Bird
Flare Blitz
Focus Punch
Giga Impact

Altaria



Outrage
Hyper Beam
Sky Attack
Solar Beam

Scrafty



Giga Impact
Focus Punch
Foul Play
Outrage

Rhyperior



Earthquake
Focus Punch
Giga Impact
Rock Wrecker

Aggron



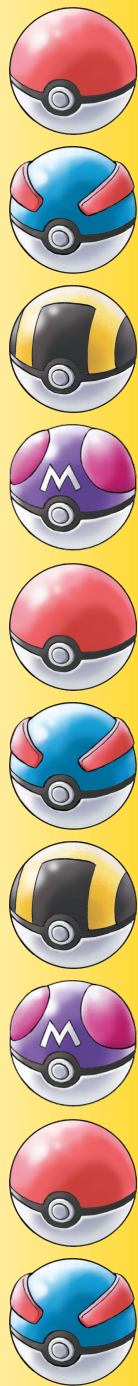
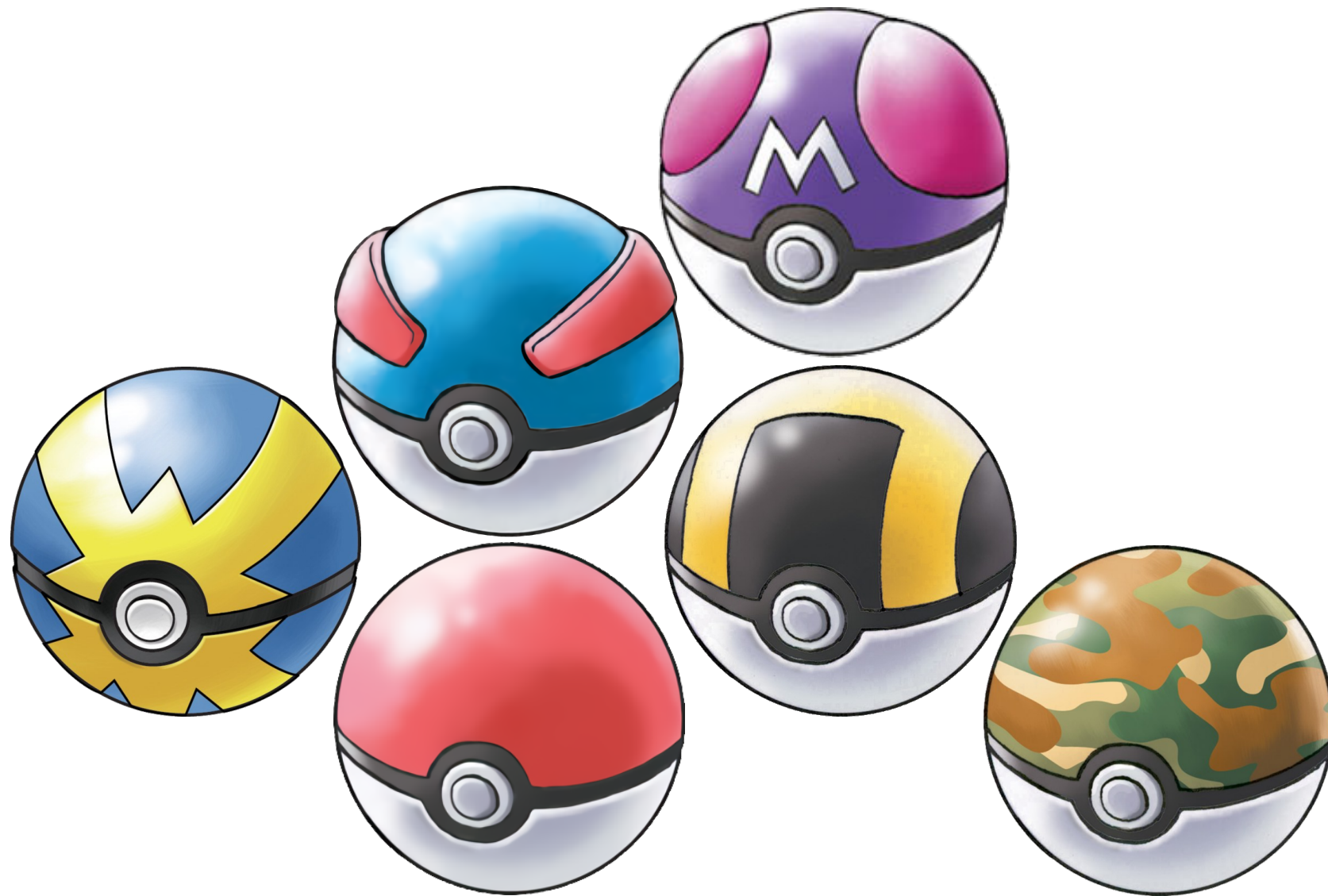
Focus Punch
Head Smash
Giga Impact
Steel Roller

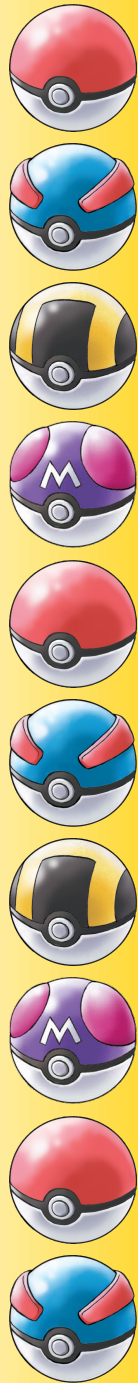
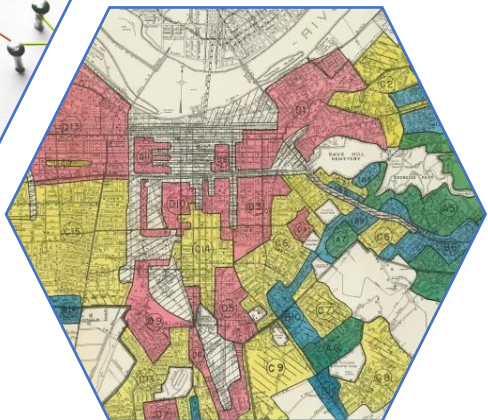
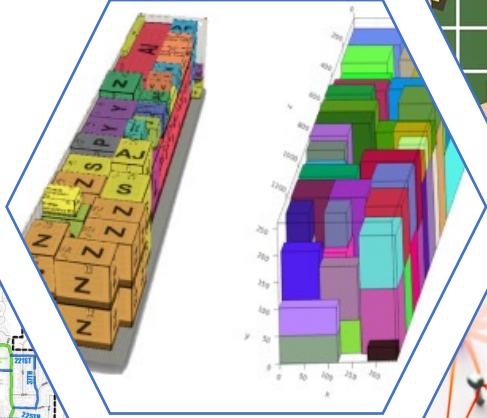
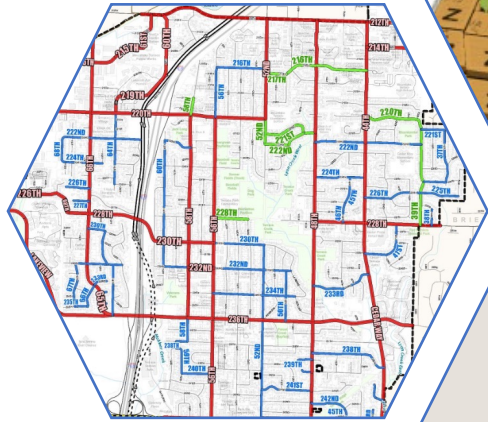
Turtonator



Draco Meteor
Hyper Beam
Shell Trap
Solar Beam







Multi-Objective Linear Programming to find the Best Pokémon Team

Dr Geraint Palmer
@GeraintPalmer



github.com/geraintpalmer/optimeiddio_timau_pokemon