Power and Mobility Optimisation in a Multi-Agent Electronic Warfare Game

Rachel Shaw, Joerg Fliege

Power and Mobility Optimisation in a Multi-Agent Electronic Warfare Game

Rachel Shaw, Joerg Fliege

Operational Research Group
University of Southampton

July 2025

Table of Contents

Table of Contents

Problem Formulation

Our construction consists of two fleets of UAVs: fleet F, and fleet G, where:

$$F = [a, b, c, ...] \qquad \text{and} \qquad G = [\alpha, \beta, \gamma, ...]$$

Generally, when referring to agents in either fleet, we denote $i, j \in F$ and $k, l \in G$, unless explicitly stated otherwise.

These fleets do not want to share resources - i.e. cooperate - with each other.

Communicating

Each fleet has a communication graph:

$$E(F) \subset F \times F \qquad \text{and} \qquad E(G) \subset G \times G$$

Then, if $(a, b) \in E(F)$, then we say that UAV a is communicating with UAV b.

$\implies$ Similarly, if $(\alpha, \beta) \in E(G)$, then we say that UAV α is communicating with UAV β .

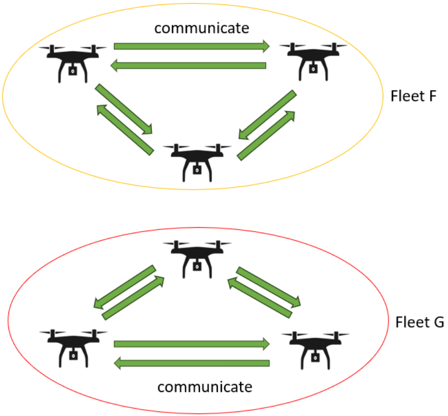

Figure: Ex.: Three drones in each fleet. Communication edges are green arrows.

Jamming

As well as communicating, the UAVs have the ability to disrupt the communication channel of the opposing fleet; we call this jamming.

This problem is modelled as a Nash game, where both players control their fleets.

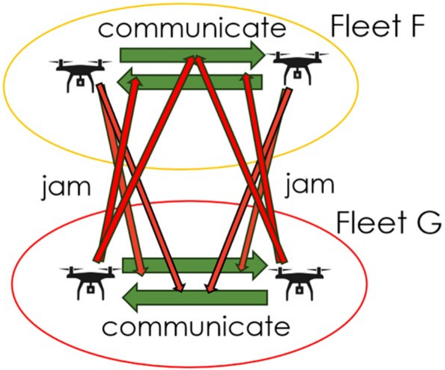

Figure: Ex.: Two drones in each fleet. Communication edges are green arrows. Opportunities for jamming are in red. Each arrow represents one decision variable.

Communication Power

Let $e := (i, j) \in E(F)$.

If UAV i uses energy $p_{i,j}$ [Watt] to send information to UAV j, then j receives an amount of power equal to

$$p^R_{i,j} = \rho \cdot p_{i,j} \cdot d_{i,j}^{-\alpha},$$

where:

- $d_{i,j} = \|x_i - x_j\|$: The euclidean distance between UAV i and j.
- ρ: 'Antenna gain'; a constant dependent on antenna design.
- α: 'Pathloss coefficient'; a coefficient dependent on environment.

Signal-to-noise-and-interference Ratio

A important quantity for the edge $e = (i, j)$ is the signal-to-noise-and-interference ratio (SINR),

This is the **payoff** function in the game.

$$SINR_{i,j} = \frac{p^R_{i,j}}{\sigma^2 + I_{(i,j)}}$$

where,

- $p^R_{i,j}$: The power received by agent j from agent i.
- $\sigma^2 > 0$: 'Noise'; a very small constant.
- $I_{(i,j)}$: The interference due to jamming from fleet G.

Importance of SINR

Why is the SINR important?

- Engineers want to compute the bit error rate (BER), a parameter indicating the quality and reliability of the data transfer of a channel.
- We can use the SINR to compute this:

$$\text{BER} = \frac{1}{2} \cdot \text{erfc}\left(\sqrt{\frac{\text{SINR} \cdot B}{f_b}}\right),$$

where:

- erfc: The complementary error function,
- B: Bandwidth of the channel (in Hz),
- f_b: Data rate or bit rate (in bits per second).

Interference Power

$$SINR_{i,j} = \frac{p^R_{i,j}}{\sigma^2 + I_{(i,j)}}$$

How do we compute the total interference?

$$I_{(i,j)} = \rho \sum_{k \notin F} p^J_{k,(i,j)} d_{k,j}^{-\alpha},$$

where:

- ρ: 'Antenna gain'; a constant dependent on antenna design.
- $d_{k,j} = \|x_k - x_j\|$: The Euclidean distance between UAV k and j.
- $p^J_{k,(i,j)}$: Power sent from UAV k in the opposing team to disrupt communication along edge (i,j).

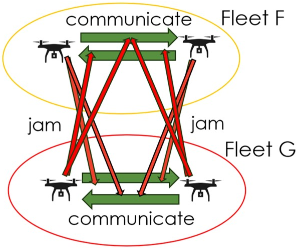

Figure: Power sent to the opposing fleet is aimed at the communication edge, not the agent itself.

Fleet F's Objectives

Let's look at the problem from fleet F's point of view.

- Fleet F wants to maximise their SINR:

$$\max_{p_{i,j},x_i,x_j} \quad f_1 = \sum_{\substack{i,j\in F\\ i\neq j}} \frac{\rho p_{i,j} d_{i,j}^{-\alpha}}{\sigma^2 + \rho \sum_{k\notin F} p^J_{k,(i,j)} d_{k,j}^{-\alpha}}$$

Fleet F's Objectives

Let's look at the problem from fleet F's point of view.

- Fleet F wants to maximise their SINR:

$$\max_{p_{i,j},x_i,x_j} \quad f_1 = \sum_{\substack{i,j\in F\\ i\neq j}} \frac{\rho p_{i,j} d_{i,j}^{-\alpha}}{\sigma^2 + \rho \sum_{k\notin F} p^J_{k,(i,j)} d_{k,j}^{-\alpha}}$$

- Whilst also minimising the opposing fleet's SINR:

$$\min_{p^J_{i,(k,l)},x_i} \quad f_2 = \sum_{\substack{k,l\notin F\\ k\neq l}} \frac{\rho p_{k,l} d_{k,l}^{-\alpha}}{\sigma^2 + \rho \sum_{i\in F} p^J_{i,(k,l)} d_{i,l}^{-\alpha}}$$

Fleet F's Objectives

Let's look at the problem from fleet F's point of view.

- Fleet F wants to maximise their SINR:

$$\max_{p_{i,j},x_i,x_j} \quad f_1 = \sum_{\substack{i,j \in F \\ i \neq j}} \frac{\rho p_{i,j} d_{i,j}^{-\alpha}}{\sigma^2 + \rho \sum_{k \notin F} p_{k,(i,j)}^J d_{k,j}^{-\alpha}}$$

- Whilst also minimising the opposing fleet's SINR:

$$\min_{p_{i,(k,l)}^J,x_i} \quad f_2 = \sum_{\substack{k,l \notin F \\ k \neq l}} \frac{\rho p_{k,l} d_{k,l}^{-\alpha}}{\sigma^2 + \rho \sum_{i \in F} p_{i,(k,l)}^J d_{i,l}^{-\alpha}}$$

- The decision variables for players in F are shown in blue, while those for players in G are shown in red. When calculating the SINR for fleet F, the decision variables of fleet G are treated as *parameters*, provided as assumed initial data.

Fleet G's Objectives

This time, let's take a brief look at the problem from fleet G's point of view.

- Similarly, fleet G wants to maximise their SINR:

$$\max_{p_{k,l}, x_k, x_l} \quad f_2 = \sum_{\substack{k,l \in G \\ k \neq l}} \frac{\rho p_{k,l} d_{k,l}^{-\alpha}}{\sigma^2 + \rho \sum_{i \notin G} p^J_{i,(k,l)} d_{i,l}^{-\alpha}}$$

- Whilst also minimising the opposing fleet's SINR:

$$\min_{p^J_{k,(i,j)}, x_k} \quad f_1 = \sum_{\substack{i,j \notin G \\ i \neq j}} \frac{\rho p_{i,j} d_{i,j}^{-\alpha}}{\sigma^2 + \rho \sum_{k \in G} p^J_{k,(i,j)} d_{k,j}^{-\alpha}}$$

Constraints for Fleet F

Constraints

- We also need to consider the constraints and bounds of this problem.
- Each agent cannot exceed some maximum power, P^{max}.

$$\sum_{\substack{j \in F \\ j \neq i}} p_{i,j} + \sum_{k,l \notin F} p^{J}_{i,(k,l)} + c_i ||x_i - x_i^0|| \leq P_i^{max}, \qquad (i \in F). \tag{1}$$

Constraints for Fleet F

Constraints

- We also need to consider the constraints and bounds of this problem.
- Each agent cannot exceed some maximum power, P^{max}.

$$\sum_{\substack{j \in F \\ j \neq i}} p_{i,j} + \sum_{k,l \notin F} p^J_{i,(k,l)} + c_i ||x_i - x_i^0|| \leq P_i^{max}, \qquad (i \in F). \tag{1}$$

- Each agent cannot exceed some δ, the maximum distance from their initial position, denoted x_i^0.

$$||x_i - x_i^0|| \leq \delta_i \qquad (i \in F) \tag{2}$$

Constraints for Fleet F

Constraints

- We also need to consider the constraints and bounds of this problem.
- Each agent cannot exceed some maximum power, P^{max}.

$$\sum_{\substack{j \in F \\ j \neq i}} p_{i,j} + \sum_{k,l \notin F} p^J_{i,(k,l)} + c_i ||x_i - x_i^0|| \leq P_i^{max}, \qquad (i \in F). \tag{1}$$

- Each agent cannot exceed some δ, the maximum distance from their initial position, denoted x_i^0.

$$||x_i - x_i^0|| \leq \delta_i \qquad (i \in F) \tag{2}$$

- Each agent must maintain some minimum distance between each other, denoted ξ, to avoid crashing.

$$||x_i - x_j|| \geq \xi \qquad (i, j \in F) \tag{3}$$

Constraints and Bounds

Bounds

- The only bounds to deal with are the non-negativity bounds associated with the power; $p_{i,j}, p^J_{i,(k,l)} \geq 0$. Provided the agents are within δ, $x_i, x_j \in \mathbb{R}^2$.

The feasible set of fleet F is denoted by C_F, and the feasible set of fleet G by C_G

Fleet F's Bi-Objective Problem

We can then write the two objectives of fleet F as one bi-objective optimisation problem:

$$\max_{\boldsymbol{x}=(p_{i,j},p^J_{i,(k,l)},x_i)} \begin{bmatrix} f_1(\boldsymbol{x},\boldsymbol{y}) \\ -f_2(\boldsymbol{x},\boldsymbol{y}) \end{bmatrix} := \begin{bmatrix} \displaystyle\sum_{\substack{i,j\in F\\ i\neq j}} \frac{\rho p_{i,j}||x_i - x_j||^{-\alpha}}{\sigma^2 + \rho\sum_{k\notin F} p^J_{k,(i,j)}||x_i - x_j||^{-\alpha}} \\ \displaystyle -\sum_{\substack{k,l\notin F\\ k\neq l}} \frac{\rho p_{k,l}||x_k - x_l||^{-\alpha}}{\sigma^2 + \rho\sum_{i\in F} p^J_{i,(k,l)}||x_i - x_l||^{-\alpha}} \end{bmatrix} \tag{4}$$

Scalarising the Problems

In order to solve both problems, we can scalarise them.

$$\max_{\boldsymbol{x}\in C_F} \quad f_1(\boldsymbol{x},\boldsymbol{y}) - f_2(\boldsymbol{x},\boldsymbol{y}) \tag{5}$$

$$\max_{\boldsymbol{y}\in C_G} \quad f_2(\boldsymbol{x},\boldsymbol{y}) - f_1(\boldsymbol{x},\boldsymbol{y}) \tag{6}$$

This construction assumes equal importance for jamming and communicating, something of which can be adjusted later.

Table of Contents

The Optimisation Problem

Rachel Shaw, Joerg Fliege

To use common optimisation techniques, we transform our *maximisation* problem, to a *minimisation* one.

We define two objectives to minimise from our initial problem:

Fleet F's problem is defined to be $\theta_F = -f_1 + f_2$

Fleet G's problem is defined to be $\theta_G = -f_2 + f_1$

The Optimisation Problem

Thus, for given parameters $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, we consider the following minimisation problem:

$$\begin{aligned} \min_{\boldsymbol{x}, \boldsymbol{y}} \quad & \theta_F(\boldsymbol{x}, \hat{\boldsymbol{y}}) + \theta_G(\hat{\boldsymbol{x}}, \boldsymbol{y}) \\ \text{s.t.} \quad & \boldsymbol{x} \in C_F, \\ & \boldsymbol{y} \in C_G \end{aligned} \tag{7}$$

Nash Equilibria of the Problem

We are looking for a Nash equilibrium $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, defined as a strategy profile where:

$$\theta_F(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq \theta_F(\boldsymbol{x}, \boldsymbol{y}^*) \qquad \forall \boldsymbol{x} \in C_F$$

$$\theta_G(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq \theta_G(\boldsymbol{x}^*, \boldsymbol{y}) \qquad \forall \boldsymbol{y} \in C_G$$

This means no player has an incentive to unilaterally change their strategy.

Convexity and Standard Approaches

If this were a convex problem, any optimum found would be a **global optimum**, thus leading to a unique Nash equilibrium.

And if this were the case, we could use the typical solution approach: reformulate as KKT, solve as a Mathematical programming with equilibrium constraints problem (MPEC).

Convexity and Standard Approaches

If this were a convex problem, any optimum found would be a **global optimum**, thus leading to a unique Nash equilibrium.

And if this were the case, we could use the typical solution approach: reformulate as KKT, solve as a Mathematical programming with equilibrium constraints problem (MPEC).

Numerical experiments show that this does not work here:

$\Longrightarrow$ θ_F and θ_G are seemingly 'too non-convex' and highly non-linear

Fixed Points & Nash Equilibria

Rosen (1965) [1] proved that a fixed point of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \mapsto (\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a Nash equilibrium for the game.

$\Longrightarrow$ Given a fixed point solution of the defined mapping, $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, that solution is a Nash equilibrium if $(\boldsymbol{x}^*, \boldsymbol{y}^*) = (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$.

This provides a theoretical foundation for iterative methods to compute equilibria.

Convexity of the problem

To circumvent the issues caused by a non-unique solution, we add a regularisation term.

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \quad & \theta_F(\boldsymbol{x},\hat{\boldsymbol{y}}) + \theta_G(\hat{\boldsymbol{x}},\boldsymbol{y}) + r\left(\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2 + \|\boldsymbol{y}-\hat{\boldsymbol{y}}\|^2\right) \\ \text{s.t.} \quad & \boldsymbol{x} \in C_F, \\ & \boldsymbol{y} \in C_G \end{aligned} \tag{8}$$

By doing this, we are restricting the new solution from straying too far from the previous solution $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$.

Splitting the Optimisation Problem

To solve this bi-objective problem, we split equation (8) into two separate problems, solving them in parallel.

This decomposition is justified by the independence of the decision variables and their associated constraints.

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \theta_F(\boldsymbol{x}, \hat{\boldsymbol{y}}) + r\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 \\ \text{s.t.} \quad & \boldsymbol{x} \in C_F \end{aligned} \tag{9}$$

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \theta_G(\hat{\boldsymbol{x}}, \boldsymbol{y}) + r\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 \\ \text{s.t.} \quad & \boldsymbol{y} \in C_G \end{aligned} \tag{10}$$

The Fixed-Point Algorithm

In order to find a fixed-point iteration for our Nash game, we utilise the following fixed-point algorithm.

Algorithm Fixed-point iteration

1: Choose $r \geq 0$ and $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$
2: Solve (9) and (10). Denote the result by $(\boldsymbol{x}, \boldsymbol{y})$
3: Set $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) := (\boldsymbol{x}, \boldsymbol{y})$
4: Update r: $r \cdot 2$
5: Repeat steps 2–4

This algorithm facilitates an iterative enhancement of strategies $\boldsymbol{x}$ and $\boldsymbol{y}$ until it finds the best response.

Fliege and Coutinho prove in their paper, [2], that this sequence converges to a global minimiser.

Table of Contents

Parameters for an Illustrative Example

Let $F = [a, b]$ and $G = [\alpha, \beta]$.

Constants:

- $\sigma^2 = 0.1$ (Noise constant)
- $\alpha = 2$ (Path-loss constant)
- $\rho = 1$ (Antenna design)

Parameters relating to constraints:

- $P_{\max} = \{a : 10.0, b : 10.0, \alpha : 10.0, \beta : 10.0\}$ (Max power)
- $\delta_i = \{a : 10.0, b : 10.0, \alpha : 10.0, \beta : 10.0\}$ (Max distance)
- $c_i = \{a : 10.0, b : 10.0, \alpha : 10.0, \beta : 10.0\}$ (Energy expenditure)

The minimum distance between two agents, ξ, is greater or equal to 1 for each agent.

Lastly, the regularisation term r is initialised to 1 and multiplied by 2 every iteration, and the tolerance for convergence is set to be 0.0001.

Plotting the UAV's path

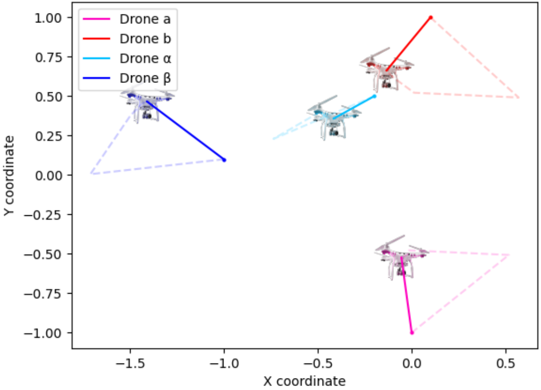

Figure: The final and initial positions, with both the shortest and convergence path.

Plotting the power distribution: pie chart

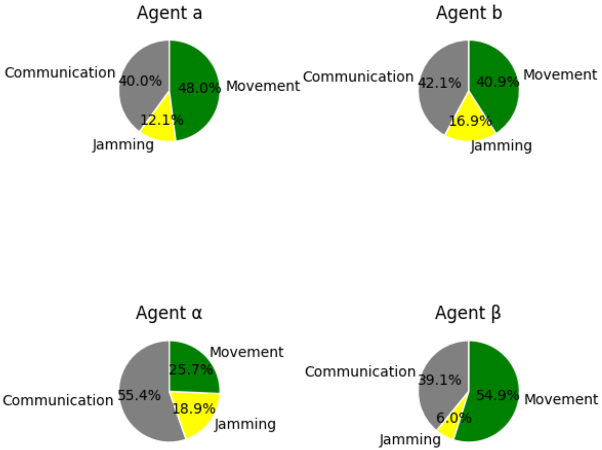

Figure: A pie chart indicating how much power for each agent is being allocated to a certain action.

Power allocation table

	To a	To b	To α	To β	To (a, b)	To (b, a)	To (α, β)	To (β, α)
a	-	4.00	-	-	-	-	1.21	0.00
b	4.21	-	-	-	-	-	1.00	0.69
α	-	-	-	5.54	0.77	1.12	-	-
β	-	-	3.94	-	0.15	0.45	-	-

Table: Power Allocation Table

	Power to Move	Total Power Used	Distance Moved	$\frac{\text{Comm Power}}{\text{Distance}^2}$	$\frac{\text{Jam Power}}{\text{Distance}^2}$
a	4.80	10.0 out of 10.0	0.48 out of 10.0	17.38	5.24
b	4.09	10.0 out of 10.0	0.41 out of 10.0	25.16	10.11
α	2.57	10.0 out of 10.0	0.26 out of 10.0	83.88	28.67
β	5.49	10.0 out of 10.0	0.55 out of 10.0	12.98	2.01

Table: Power Allocation Table (continued)

Table of Contents

Conclusion

- Flexible concept for electronic warfare with drones.
- Model based on underlying physics.
- Can accommodate fleets of drones of various sizes & various objectives.
- Non-convex Nash game, but convergence established.
- Small illustrative example presented.
- Further work established and/or working on, but not mentioned:
  - Fleet and individual Preference,
  - A different preference construction (implementing strict constraints),
  - Sizing up the example,
  - Implementing fixed/ground agents.
  - Introducing uncertainty through a Bayesian game model.

Thank You For Listening.

Support by the Defence Science and Technology Laboratory (DSTL) of the UK gracefully acknowledged.

Any questions?

References I

[1] J. B. Rosen. 'Existence and Uniqueness of Equilibrium Points for Concave N-Person Games'. In: *Econometrica* 33.3 (1965). Publisher: [Wiley, Econometric Society], pp. 520–534. ISSN: 0012-9682. DOI: 10.2307/1911749. URL: https://www.jstor.org/stable/1911749 (visited on 09/10/2024).

[2] Joerg Fliege and Walton Coutinho. 'A Jamming Game for Fleets of Mobile Vehicles'. In: (2024). (Visited on 30/03/2024).

Equal communication/jamming ratio

Recall our scalarised maximisation problem:

$$\max_{\boldsymbol{x} \in C_F} \quad f_1(\boldsymbol{x}, \boldsymbol{y}) - f_2(\boldsymbol{x}, \boldsymbol{y}) \tag{11}$$

$$\max_{\boldsymbol{y} \in C_G} \quad f_2(\boldsymbol{x}, \boldsymbol{y}) - f_1(\boldsymbol{x}, \boldsymbol{y}) \tag{12}$$

In this construction, it is assumed that the weighting associated with communication and jamming weighting is the same; hence all the objective function values lie on the line $\theta_G = -\theta_F$.

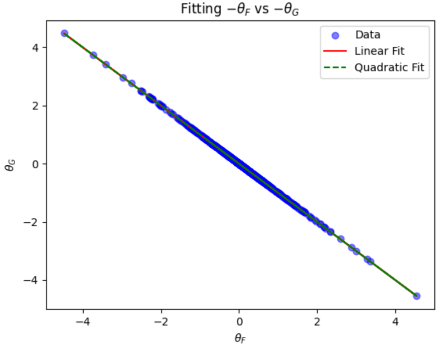

Figure: Fitted scatter-plot of θ_F by θ_G, produced by running the algorithm 1000 times, with different initial conditions.

Communication/jamming ratio

We introduce some constants $s, u \geq 0$.

Then, for fleet F, to represent the preference given to the two actions, we denote the ratio by $s : 1 - s$, where s is associated with the weighting given to communication and $1 - s$ to jamming.

We apply the same ideology to fleet G's objective function, but with u and $1 - u$.

$$\max_{\boldsymbol{x} \in C_F} \quad \theta_F(\boldsymbol{x}, \boldsymbol{y}) = s \cdot f_1(\boldsymbol{x}, \boldsymbol{y}) - (1 - s) \cdot f_2(\boldsymbol{x}, \boldsymbol{y})) \tag{13}$$

$$\max_{\boldsymbol{y} \in C_G} \quad \theta_G(\boldsymbol{x}, \boldsymbol{y}) = u \cdot f_2(\boldsymbol{x}, \boldsymbol{y}) - (1 - u) \cdot f_1(\boldsymbol{x}, \boldsymbol{y})) \tag{14}$$

Mixed Data Set

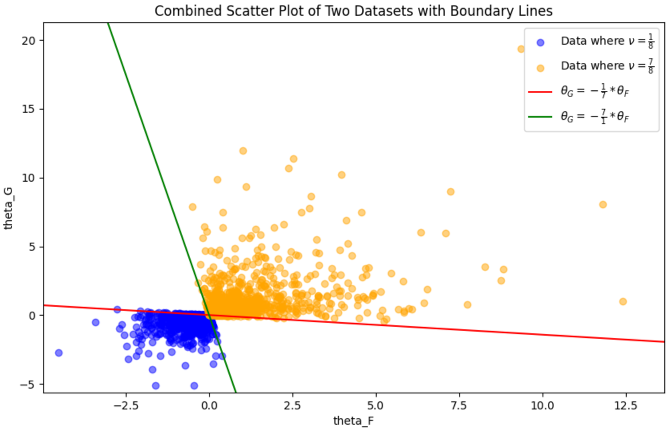

Figure: Scatter-plot of θ_F by θ_G, with data from two data sets

Proposition

Proposition

Let $(\boldsymbol{x}, \boldsymbol{y}) \in C_F \times C_G$ denote a feasible strategy profile, and define $\theta_F(\boldsymbol{x}, \boldsymbol{y})$ and $\theta_G(\boldsymbol{x}, \boldsymbol{y})$ as the associated objective values for Fleets F and G, respectively. Then, depending on the relationship between the parameters s and u, the set of admissible objective value pairs (θ_F, θ_G) induced by feasible strategies satisfies:

1. If $s > 1 - u$, then all feasible (θ_F, θ_G) satisfy
$$\theta_G(\boldsymbol{x}, \boldsymbol{y}) \geq -\frac{u}{1-s} \cdot \theta_F(\boldsymbol{x}, \boldsymbol{y}) \quad \text{and} \quad \theta_G(\boldsymbol{x}, \boldsymbol{y}) \geq -\frac{1-u}{s} \cdot \theta_F(\boldsymbol{x}, \boldsymbol{y}),$$
2. If $s = 1 - u$, then all feasible strategy profiles satisfy $\theta_G(\boldsymbol{x}, \boldsymbol{y}) = -\theta_F(\boldsymbol{x}, \boldsymbol{y})$,
3. If $s < 1 - u$, then all feasible (θ_F, θ_G) satisfy
$$\theta_G(\boldsymbol{x}, \boldsymbol{y}) \leq -\frac{u}{1-s} \cdot \theta_F(\boldsymbol{x}, \boldsymbol{y}) \quad \text{and} \quad \theta_G(\boldsymbol{x}, \boldsymbol{y}) \leq -\frac{1-u}{s} \cdot \theta_F(\boldsymbol{x}, \boldsymbol{y}).$$