数学科と連携した情報の授業実践について

令和4年度全国高等学校情報教育研究会
東京都立新宿山吹高等学校
中山　享司

これからお話しすること

- 授業計画
- 数理最適化とは
- こんなところで使われています
- （計画に即して）問題
- 数学科との連携
- アプリ作ってみました
- 結びの言葉

- 授業計画
- 数理最適化とは
- こんなところで使われています
- （計画に即して）問題
- 数学科との連携
- アプリ作ってみました
- 結びの言葉

指導計画について

（1）情報社会の問題解決

（ア）問題を発見・解決する方法	4
（イ）法・情報セキュリティ・情報モラル	4
（ウ）情報技術と情報社会	4

（2）コミュニケーションと情報デザイン

（ア）メディアとコミュニケーション	4
（イ）情報デザインと役割	2
（ウ）コミュニケーションと情報デザイン	6

（できるだけ）毎回5～10分程度、
プログラムの指導に充てたい
→（3）の指導の効率化を目指す

（3）コンピュータとプログラミング

（ア）コンピュータのしくみと処理	3
（イ）アルゴリズムとプログラム	8
（ウ）モデル化とシミュレーション	5

（4）情報通信ネットワークとデータの活用

（ア）ネットワークのしくみと構成要素	4
（イ）データベースの仕組みと活用	5
（ウ）データの収集と傾向の可視化	7

ここでの実施を想定

「情報Ⅰ」ミニマムモデルより

授業計画

- 全３時間を（ひとまず）想定

時間	内容	ねらい
1	数理最適化とは？線形計画法の紹介、実装	少ない変数の題材をモデル化できるようにする
2	複数の式での解法、ナップサック問題	制約条件が複雑になった題材をモデル化できる 題材をもとに、自らモデルを構築できるようになる
3	外部ファイルを使った解法（主婦問題）	コンピュータの特性を活かして、条件が多数ある題材でもモデルを構築することができるようになる

- 授業計画
- **数理最適化とは**
- こんなところで使われています
- （計画に即して）問題
- 数学科との連携
- アプリ作ってみました
- 結びの言葉

数理最適化とは

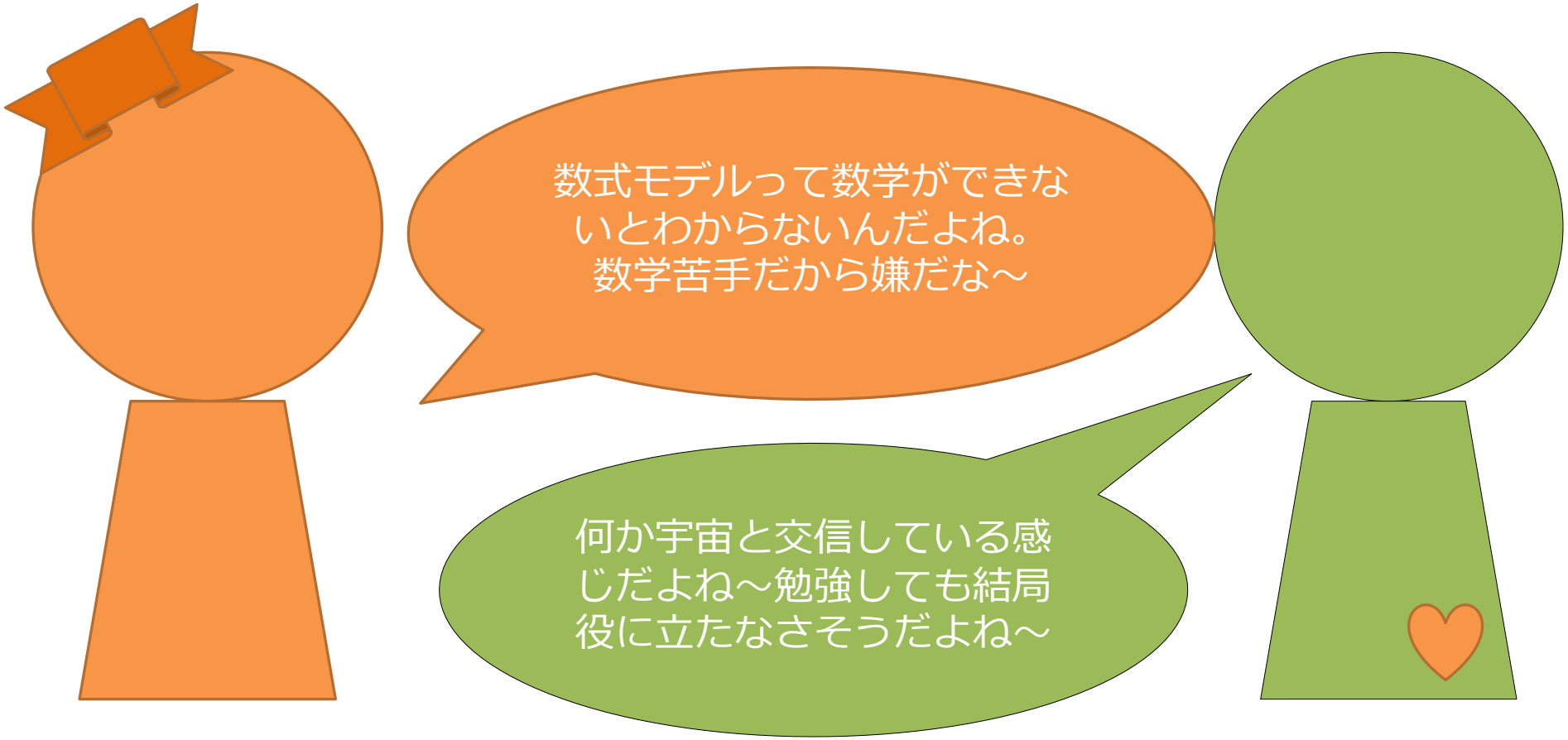
- 現実の問題を数式（数理モデル）として定義
- 制約条件を満たしつつ
- コストの最小化や利益が最大化されるような変数の値を求める手法

数式で表したものを目的関数という

こんなことに使われています

商品の配置最適化
店員のシフトスケジューリング
列車の乗務員スケジューリング
生産計画の作成　　など

こんな会話ありませんか？



この世は

- 数理最適化だらけなんです。
- 数理最適化で世の中は成り立っているんですよ！？

- 授業計画
- 数理最適化とは
- **こんなところで使われています**
- （計画に即して）問題
- 数学科との連携
- アプリ作ってみました
- 結びの言葉

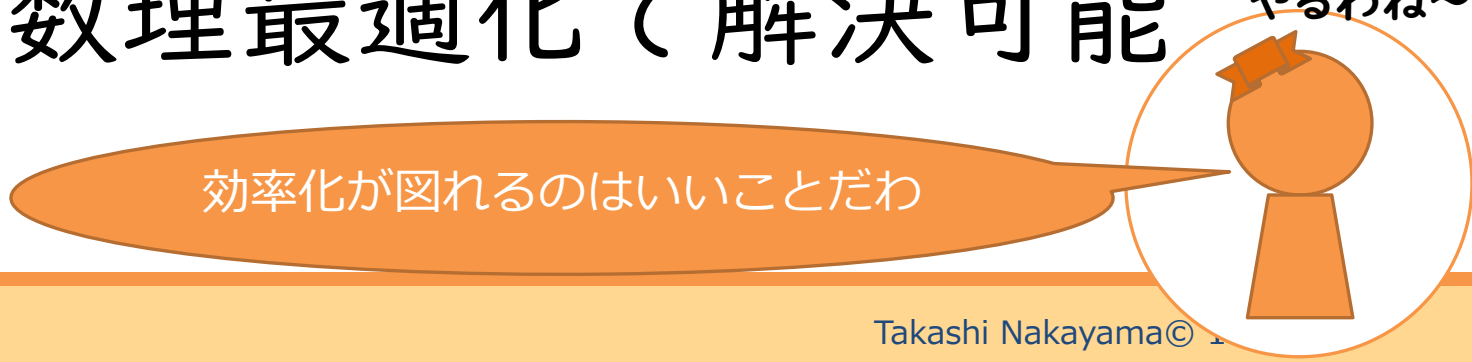
たとえば１

●宅配業者の宅配・集荷業務

車に積める荷物の量は決まっています
宅配をしながら集荷をしたいです。
（できれば）何度も同じ道を通りたくありません

どのようなルートが一番効率がいいでしょうか？

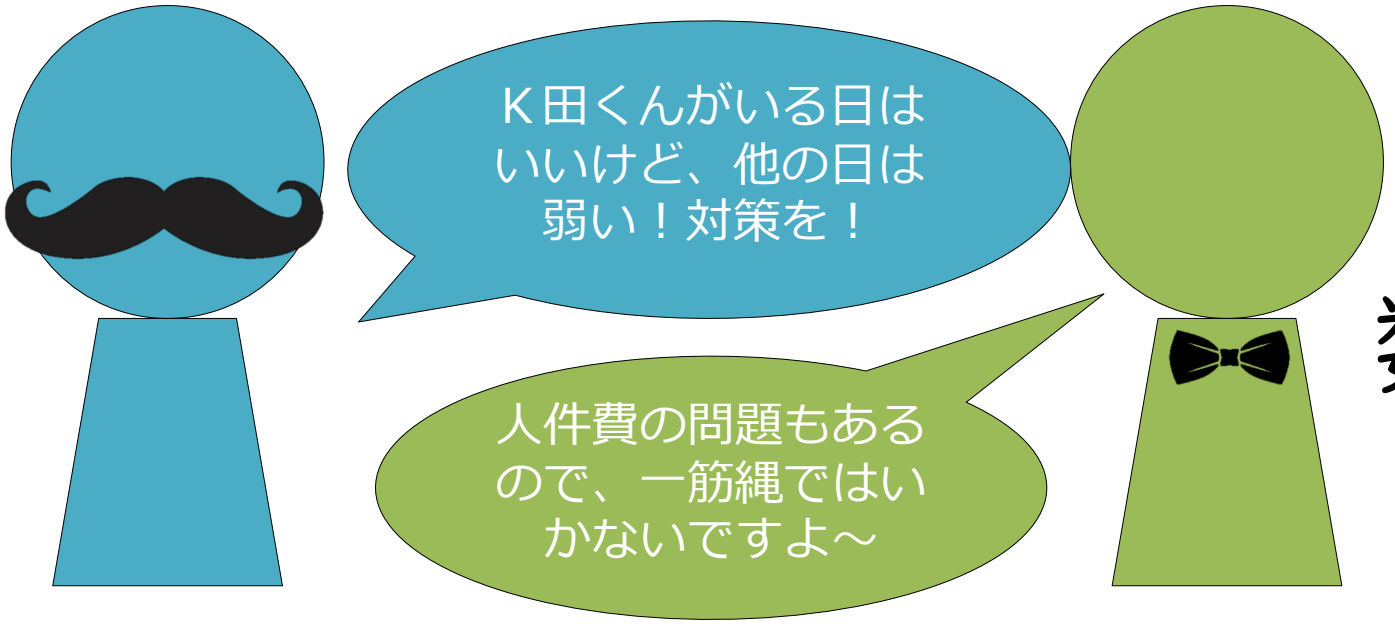
数理最適化で解決可能



たとえば２

●アルバイトのシフト問題

従業員ごとに、出勤できる曜日・時間が決まっている
従業員ごとに、専門の業務がある
その日ごとに、必要な従業員数が違う



数理最適化で解決可能

https://www.msiism.jp/article/jrs-shift-scheduling.html

たとえば3

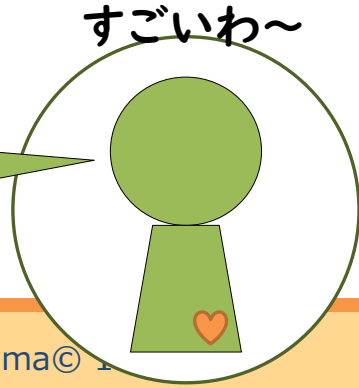
●配車計画

週末はなかなかタクシーがつかまらない
呼んでも待ち時間が長い

数理最適化で解決可能

配車の最適化（待ち時間を全体で最適・最短にする）

需要予測（機械学習）もしている



便利なサービスの中には数理最適化があるのね

- 授業計画
- 数理最適化とは
- こんなところで使われています
- **（計画に即して）問題**
- 数学科との連携
- アプリ作ってみました
- 結びの言葉

例題

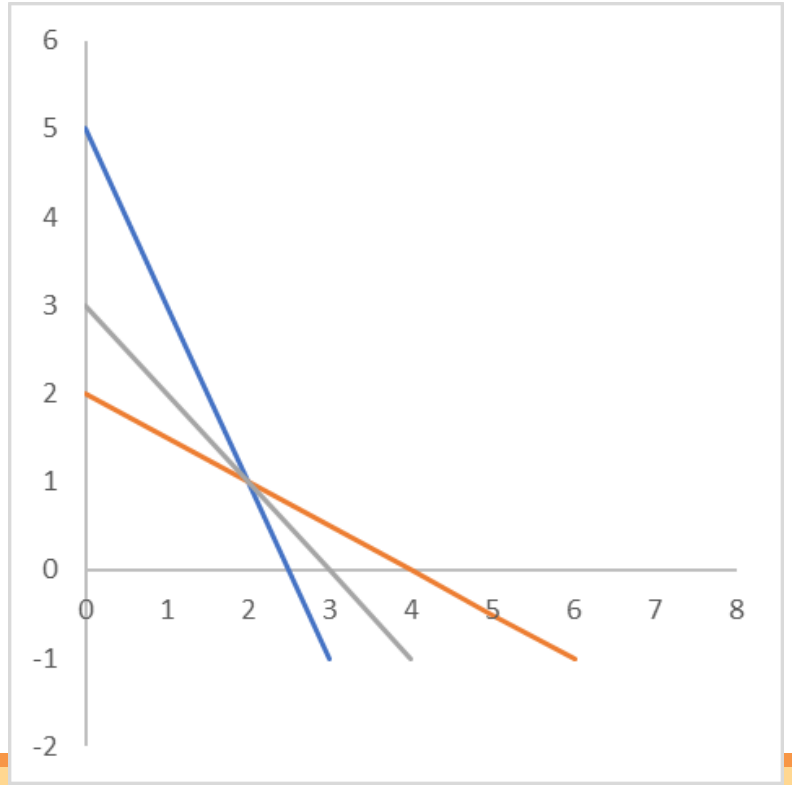
- x+yが最大となるxとyのそれぞれ浮動小数点と整数の組み合わせを求めよ

＜条件＞

$x \geqq 0, y \geqq 0$

$2x+y \leqq 5$

$x+2y \leqq 4$



まずは

解答例

```
!pip install pulp
from pulp import *
```

ライブラリのインストール
（グーグルドライブは必須）

pulpというライブラリに入っている関数
やクラスを全て使える

プログラムの実装

解答例

```
problem = LpProblem('simple_problem', LpMaximize)
x = LpVariable('x')
y = LpVariable('y')
problem += x + y
problem += -1*x + 3 >= y
problem += -1/2*x +2  >= y
```

問題の定義

変数の定義
カンマで区切って、
cat="Integer"と入れると
整数で定義される

目的関数（これを最大化したい）

制約条件の定義

右辺のyに係数がつくとエラー

プログラムの実装

解答例

- status = problem.solve()
- print(LpStatus[status])
- print("x:",x.value())
- print("y:",y.value())

解けたらoptimal（最適）と表示

このコード自体は使いまわせば良い

結果の表示

指導のポイント

- コードは
  - 問題の定義
  - 変数の定義
  - 目的関数の定義
  - 制約条件の定義　　・・・を記述する。

何を最大化（最小化）したいのか
（目的関数は何か）と
制約条件は何かを考えさせる
＝定式化

- 書き方として・・・
  - 目的関数は
    problem ＋＝ x ＋ y　（式１）　となっているのに対して
  - 制約条件は
    problem ＋＝ -1*x ＋ 3　（式１）　>＝ y　（式２（片方は定数でも良い））　となっているところに注意

等式でつなぐ

指導のポイント

- なんで線形計画法をやるんですか？

→身の回りのことを抽象化（モデル化）できるようにする

- この時にモデル化された数式の解法が線形計画法（基礎的なので取っ掛かりやすい）
- 意外とたくさんのことが解決できる！
- どのように抽象化したら良いのかを検討
- 基本的な手法で広範囲の問題を解決できる

食事問題

問題）食品A,B,Cの３種類を組み合わせて摂取する。

栄養素摂取量を満たした上で，食費を最小限にできる摂取量を求める。

●食品

・食品A：価格=20, 栄養素1=22, 栄養素2=20, 栄養素3=10

・食品B：価格=12, 栄養素1=13, 栄養素2=30, 栄養素3=5

・食品C：価格=18, 栄養素1=17, 栄養素2=5, 栄養素3=12

●必要な栄養素摂取量

・栄養素１：200

・栄養素２：200

・栄養素３：100

https://www.y-shinno.com/pulp-intro/より

プログラムの実装

解答例

```
problem = LpProblem(  'syokuji', LpMinimize)
A = LpVariable(name = "A", lowBound = 0, cat="Integer")
B = LpVariable(name = "B", lowBound = 0, cat="Integer")
C = LpVariable(name = "C", lowBound = 0, cat="Integer")

problem += 20*A + 12*B + 18*C

problem += 22*A + 13*B + 17*C >= 200
problem += 20*A + 30*B + 5*C >= 200
problem += 10*A + 5*B + 12*C >= 100
```

価格の最小値を求める

lowBound 変数の下限値

catで変数の種類を宣言。Integer＝整数

ここを隠して、目的関数と制約条件を考えさせる

ナップサック問題

問題）様々な冒険を経て宝島に辿り着いたあなたは、そこにある全ての宝をリュックに入れて持ち帰ろうとしました。しかし残念ながら、リュックには65kgまでしか宝を詰めることが出来ないため、うまくお宝を選びながら利益が最大になるような組み合わせを考える必要があります。
ここではそれぞれのお宝は無限にあるとします

宝島のお宝

宝0：value0 ＝ 105万円 / weight0 ＝ 10kg
宝1：value1 ＝ 140万円 / weight1 ＝ 13kg
宝2：value2 ＝ 65万円 / weight2 ＝ 6kg
宝3：value3 ＝ 90万円 / weight3 ＝ 11kg
宝4：value4 ＝ 220万円 / weight4 ＝ 20kg
宝5：value5 ＝ 170万円 / weight5 ＝ 18kg

利益が最大になるように、
お宝をリュックに詰めたい！

指導のポイント

- 目的関数、制約条件が問題文だけだと分かりづらい

一言で言えば利益（＝価値）

- 目的関数：お宝の価値×お宝を持っていく個数
（これを最大化したい）

一言で言えば重量

これの存在に気づかせる

- 制約条件：お宝の重さ×お宝を持っていく個数

解答例

formatメソッドにより、amount というリスト全体を初期設定

- values = [105,140,65,90,220,170]
- weights = [10,13,6,11,20,18]
- problem = LpProblem('knapsack', LpMaximize)

リストの要素数
今回は0～5のループ

- amounts = [LpVariable('amount_{0}'.format(i), lowBound = 0, cat = "Integer") for i in range(len(weights))]
- problem += lpDot(values, amounts)
- problem += lpDot(weights, amounts) <= 65
- print(problem)

右の（）内の内積

formatメソッドについては　https://www.javadrive.jp/python/string/index24.html

動かしたら

- コンピュータを使うと、
一瞬であらゆる組み合わせを試行することができる
（そういうプログラムを作る）
- 数字を変えてやり直すことも簡単に可能
すぐに解が求まる

モデルを作るメリット

- 身の回りのことを題材にして、問題を作らせる
（スマホの契約で、最低料金になるプランの検討など）

主婦問題

問題）food.csv にリストアップされている７７種類の食料品から何をどのくらい購入すれば、nutrient.csv に記載された１日に健康維持に必要な栄養素を最安で充足することができるか？

food.csvの一部

food	Calories	Protein	Calcium	Iron	Vitamin A	Thiamine	Riboflavin	Niacin	Ascorbic Acid (Vitamin C)		
WheatFlo	44.7	1411	2	365	0	55.4	33.3	441	0		
Macaroni	11.6	418	0.7	54	0	3.2	1.9	68	0		
WheatCer	11.8	377	14.4	175	0	14.4	8.8	114	0		
Corn Flake	11.4	252	0.1	56	0	13.5	2.3	68	0		
Corn Mea	36	897	1.7	99	30.9	17.4	7.9	106	0		
Hominy G	28.6	680	0.8	80	0	10.6	1.6	110	0		
Rice	21.2	460	0.6	41	0	2	4.8	60	0		
Rolled Oat	25.3	907	5.1	341	0	37.1	8.9	64	0		
White Brea	15	488	2.5	115	0	13.8	8.5	126	0		
Whole Wh	12.2	484	2.7	125	0	13.9	6.4	160	0		
Rye Bread	12.4	439	1.1	82	0	9.9	3	66	0		
Pound Cal	8	130	0.4	31	18.9	2.8	3	17	0		
Soda Crac	12.5	288	0.5	50	0	0	0	0	0		
Milk	6.1	310	10.5	18	16.8	4	16	7	177		
Evaporate	8.4	422	15.1	9	26	3	23.5	11	60		
Butter	10.8	9	0.2	3	44.2	0	0.2	2	0		
Oleomarga	20.6	17	0.6	6	55.8	0.2	0	0	0		
Eggs	2.9	238	1	52	18.6	2.8	6.5	1	0		
Cheese (C	7.4	448	16.4	19	28.1	0.8	10.3	4	0		
Cream	3.5	49	1.7	3	16.9	0.6	2.5	0	17		
PeanutBu	15.7	661	1	48	0	9.6	8.1	471	0		
Mayonnais	8.6	18	0.2	8	2.7	0.4	0.5	0	0		

nutrient.csv

Nutrient	Intake	Unit
Calories	3	kilocalories
Protein	70	grams
Calcium	0.8	grams
Iron	12	milligrams
Vitamin A	5	10001U
Thiamine	1.8	milligrams
Riboflavin	2.7	milligrams
Niacin	18	milligrams
Ascorbic A	75	milligrams

解答例

- from pulp import *
- import numpy as np
- import pandas as pd

ライブラリを一通り準備

csv読み込み

- df_food = pd.read_csv(’food.csv’)
- df_nutrient = pd.read_csv(’nutrient.csv’)

解答例

- problem = LpProblem(‘syufu’ ,LpMinimize)

問題の定義（最小化）

- food_amount = [LpVariable('food_{0}'.format(i), lowBound=0) for i in range(len(df_foods))]

変数の定義

- problem += lpSum(food_amount)

目的関数

- for k,v in df_nutrient.iterrows():

  problem += lpDot(food_amount,df_foods[v.Nutrient]) >= v.Intake

制約条件

例えばこんな問題が作れます

（現代にアレンジ）スムージー問題

- 栄養とりたい
- カロリー摂りたくない

- バナナ・トマト・きゅうり・・・・・の栄養表を用意
- どれを組み合わせて、どんな商品にする？

こんな問題どうでしょう

指導のポイント

- 問題（題材）を定式化するイメージが持てるか。
- 文章題の文書を読んで、式にすることができるか

- 授業計画
- 数理最適化とは
- こんなところで使われています
- （計画に即して）問題
- **数学科との連携**
- アプリ作ってみました
- 結びの言葉

数学科との連携

- ある題材について・・・
- 数学科でやること
- 情報科でやること
- 教材を共有し検討する

〇〇は数学
■■は情報
と一概に言えない

- 有機的な連携を図り、問題解決のための資質・能力を育成！

異なる役割を持ったものたちが組織全体として
１つの目標を達成するために相互作用

- 授業計画
- 数理最適化とは
- こんなところで使われています
- （計画に即して）問題
- 数学科との連携
- **アプリ作ってみました**
- 結びの言葉

ここで

- アプリを作りました
- 会計時に硬貨をを
  ・なるべく少ない数で支払う
  ・なるべく多くの数で支払う
  これらがシミュレーションできるアプリ

- https://coin-problem.herokuapp.com

- 授業計画
- 数理最適化とは
- こんなところで使われています
- （計画に即して）問題
- 数学科との連携
- アプリ作ってみました
- **結びの言葉**

- 最初は３時間の指導計画を構想していたが、「自分で問題を作らせる」という活動を入れると、５時間くらい欲しい

身の回りのことを抽象化（モデル化）して、問題解決する力は、コンピュータがこれだけ普及した現代においては、必要な力！