

# 集合値プログラミング

## 第5章 最適化(3)

06T4056N 三沢博章

# はじめに

- 嗜好への最適化

→有限な資源を人の嗜好によって配分する方法

- ネットワークの可視化

→ネットワークを可視化する際に交点を最小限にする

# 嗜好への最適化

- 学寮の最適化

- 5つの寮のそれぞれに2人分のスペースがある
- 10人の学生から第1希望、第2希望をとる
- 誰かしら第1希望or第2希望には入れない

寮={'Zeus','Athena','Hercules','Bacchus','Pluto'}

人=Toby 第1希望:Bacchus 第2希望:Hercules

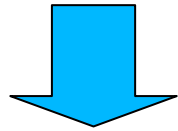
Steve 第1希望:Zeus 第2希望:Pluto

Andrea 第1希望:Athena 第2希望:Zeus

etc...

# 解の表現

- 有効な解を見つける
  - 必ずしも優れた解でなくてもよい
  - 各寮にちょうど2人の学生が割り当てられる



- 各寮に2つのスロット
- 各学生を順々にオープンなスロットに割り当てる

# コスト関数

- ① スロットのリストを生成
- ② 寮を割り当てる
- ③ コストを計算する
- ④ スロットを削除

{ 第1希望に割り当て・・・コスト 0  
第2希望に割り当て・・・コスト 1  
どちらでもない     ・・・コスト 3

解の表現、コスト関数が揃うと



最適化が実行できる

# ネットワークの可視化

- ネットワークの可視化とは？
  - 人々とその間のリンクを可視化する
  - 可視化することで人々のコネクションを発見する
- ソーシャルネットワーク …… 人と人との接続  
例 : MySpace、Facebook、LinkedIn、mixi

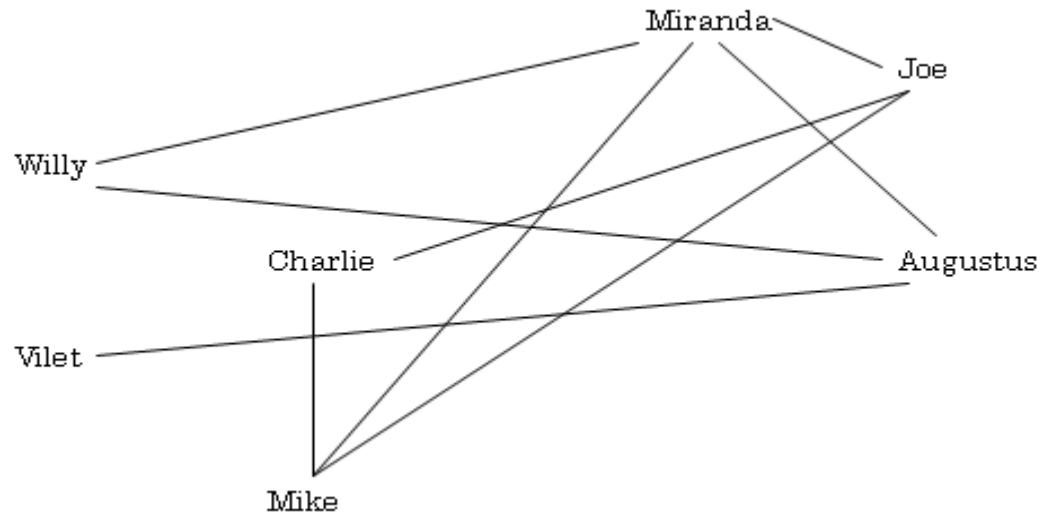
# レイアウト問題

- 名前の配置

→ 適当に配置すると交点が増え混乱しやすい

故に、誰は誰の友達かという事実のリストを取り、解釈の容易なネットワーク図を生成するプログラムを作成

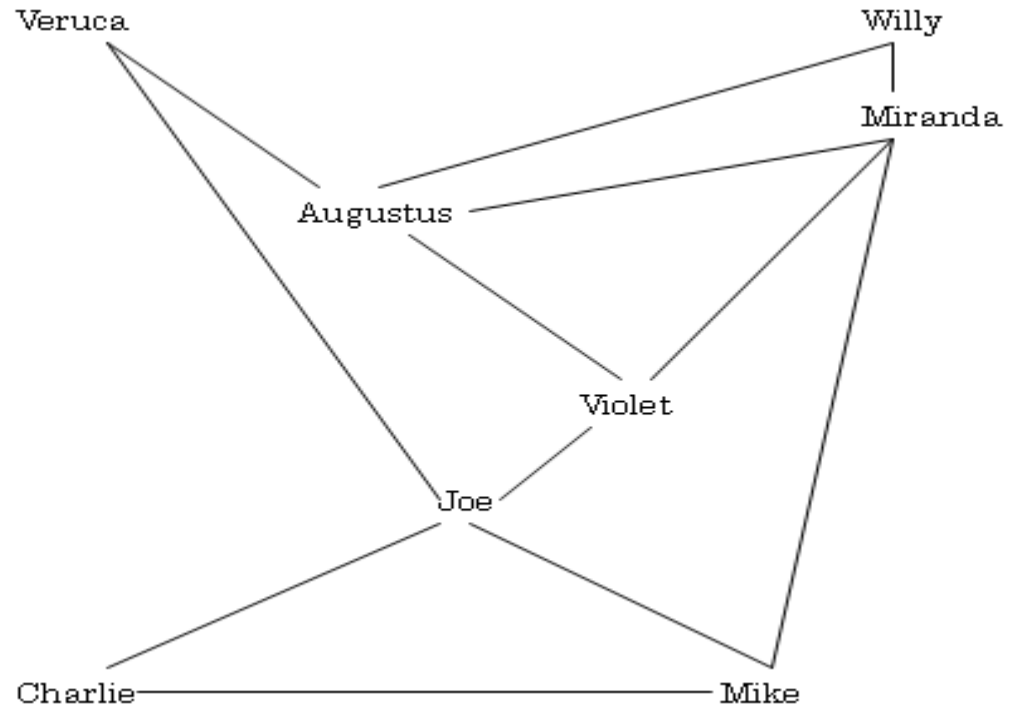
悪い例:



# レイアウト問題(2)

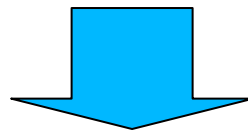
- 最適化を使ったレイアウト生成
  - コスト関数を決定し、最小化をする
  - 交差した線の数を数えるコスト関数

実行例:



# さらなる可能性

- 最適化アルゴリズムの重要なステップ
  - 解の表現
  - コスト関数の決定



これが可能ならば、、、

身近な問題でも最適化が適用可能な見込みが高い

例: チーム分け(最適化によりスキルが等しく配分されたチーム)