

2016/5/20 Fri.

問題解決技法入門

2. 社会的選択: 安定性
安定結婚問題

堀田 徹介

浮気しない？カップル

- 6人の男女がいます。少子化対策？のため、6組のカップルを作り結婚させちゃいましょう。でも各自の好き嫌いを考えずに強引にくっつけちゃうと、浮気する人が出るかもしれません。浮気しないように6組のカップルをつくれますか？



どうすれば浮気しないの？

浮気しないってどういうこと？

浮気ってどういう状況で起こる？

浮気する・しないを「上手く定義」する

問題解決

「問題の把握」から「意思決定」までの流れ

問題

モデル化

解く

解釈・評価

提案・解決

問題・目的の明確化

問題発見・状況認識

- 状況を把握し問題の背後にある本質を追究
- いったい何を知りたいのか？
- 問題の本質は何か？

代替案立案
モデル構築

推論・モデル作成

- 推論に基づきモデル作成
- 現実を支配する法則を数量的に明確化

答えを導く

- 解法選択
- 解法構築
- パラメータ調整

結果の解釈・評価
代替案評価・選択

結果評価・解釈

- 解法のもたらす結果の解釈・考察
- 得られた代替案の評価・分析

意思決定

論理的思考力
データ分析, 統計学
数理的アプローチ

説得力
問題解決力

問題の見直し
問題の本質を再考

モデルの妥当性評価
現実との乖離の検証

浮気する・しないを「上手く定義」する

現状認識力
問題発見・定義

安定結婚問題

- n 人の男性の集合と、 m 人の女性の集合が存在し、各人は異性全員の選好順序をもっている。このとき、安定なマッチングを見つけない。

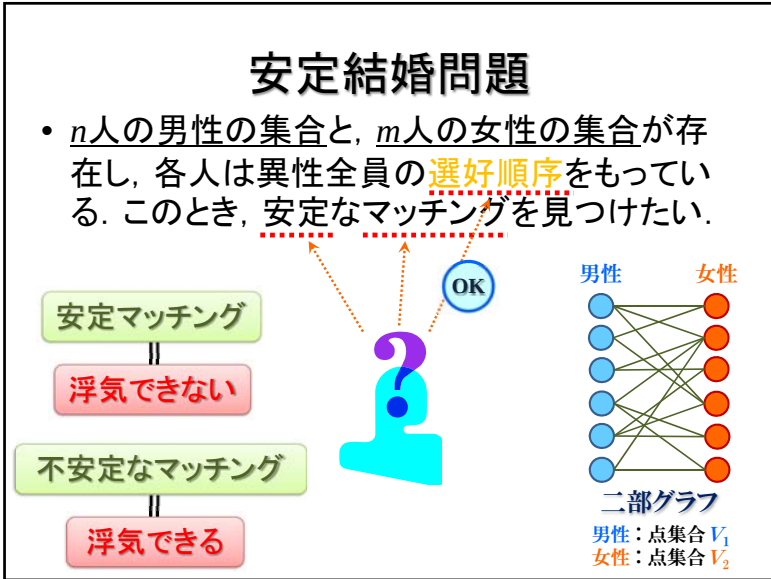
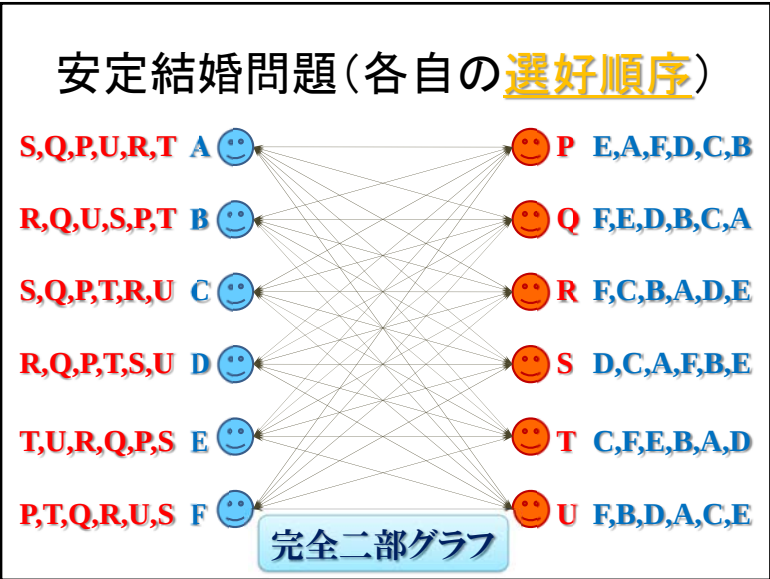
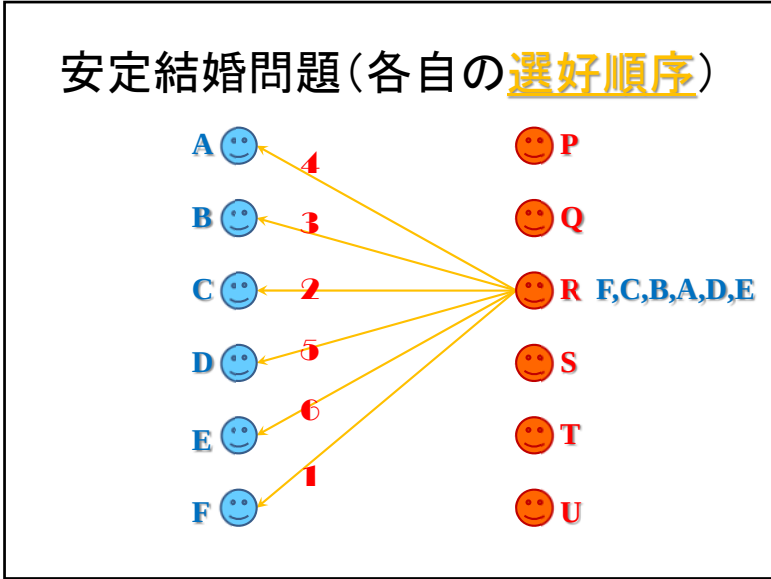
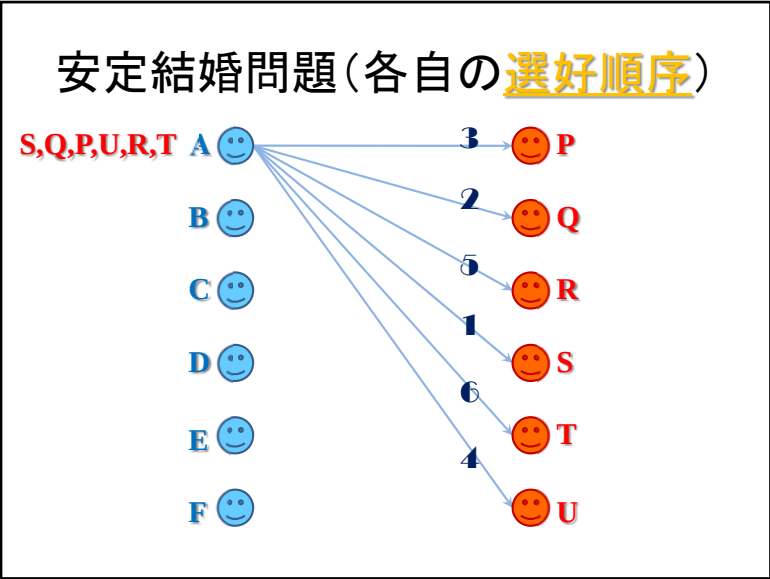
安定マッチング

浮気できない

不安定なマッチング

浮気できる





安定結婚問題(マッチング)

A

B

C

D

E

F

P

Q

R

S

T

U

マッチング

端点を共有しない枝の集合

つまり、どの点(node)も高々1本の枝(edge)にのみ接続(incident to)している

完全マッチング

全ての点(node)が、マッチング(matching)の枝(edge)に接続しているとき、そのマッチングを完全マッチングという

安定結婚問題(マッチング)

A

B

C

D

E

F

P

Q

R

S

T

U

この枝集合は、マッチング(matching)ではない

なぜかわかる？

その通り！ マッチングではありません。

なぜなら、端点を共有する枝がある(二股をかけている人がいる)から

安定結婚問題(マッチング)

A

B

C

D

E

F

P

Q

R

S

T

U

この枝集合は、マッチング(matching)だろうか？

マッチング(matching)です。でも、完全マッチング(perfect matching)ではないので、ペアを組んでない人がいるね。

つまり、我々は完全マッチングを求めたいのだよ

※男女が同数でない場合は、完全マッチング(perfect matching)は存在しないので、最大マッチング(maximum matching)を求めます。

安定結婚問題

• n人の男性の集合と、m人の女性の集合が存在し、各人は異性全員の選好順序をもっている。このとき、安定なマッチングを見つけない。

安定マッチング

浮気できない

不安定なマッチング

浮気できる

OK

OK

?

Confidential

浮気する(不安定な)カップルとは？

こんな2組のカップル(マッチング)を作ってしまったら...

このマッチングは**不安定**！
なぜなら
ブロッキング・ペア
が存在するから！

そんな～ ひどいわ

浮気

浮気しない(安定な)恋人たち

浮気しない(できない)恋人たち

このマッチングは**安定**！
なぜなら
ブロッキング・ペア
が存在しないから

浮気を試みるも... (´ω´)

拒絶 誘い

安定結婚問題

- n 人の男性の集合と, m 人の女性の集合が存在し, 各人は異性全員の**選好順序**をもっている. このとき, **安定なマッチング**を見つけない.

安定マッチング
||
浮気できない
不安定なマッチング
||
浮気できる

安定結婚問題(まとめ)

浮気しないカップルをつくる(安定結婚問題を解く)ということは,
(ブロッキング・ペアが存在しない)安定な完全マッチングを求めること

※男女が同数でない場合は, 完全マッチング(perfect matching)は存在しないので, 最大マッチング(maximum matching)を求めます.

問題: このマッチングは安定?

S, Q, P, U, R, T

A

😊

→

P

E, A, F, D, C, B

😊

R, Q, U, S, P, T

B

😊

→

Q

F, E, D, B, C, A

😊

S, Q, P, T, R, U

C

😊

→

R

F, C, B, A, D, E

😊

R, Q, P, T, S, U

D

😊

→

S

D, C, A, F, B, E

😊

T, U, R, Q, P, S

E

😊

→

T

C, F, E, B, A, D

😊

P, T, Q, R, U, S

F

😊

→

U

F, B, D, A, C, E

😊

問題解決

「問題の把握」から「意思決定」までの流れ

問題

→

モデル化

→

解く

→

解釈・評価

→

提案・解決

問題・目的の明確化

代替案のモデル化

結果の解釈・評価
案評価・選択

意思決定

問題の見直し
問題の本質を再考

モデルの妥当性評価
現実との乖離の検証

不安定マッチングを求め直す

安定マッチングを求める

問題発見・状況認識

状況把握し問題の背後に隠れた課題を明らかにする

問題の本質は何か?

安定マッチング

Q1. そんなものあるのか?

Q2. 求められるのか?

現実を支配する法則を数量的に明確化

結果評価・解釈

解法のもとで結果の解釈・考察

得られた代替案の評価・分析

演習: やってみよう

S, Q, P, U, R, T

A

😊

→

P

E, A, F, D, C, B

😊

R, Q, U, S, P, T

B

😊

→

Q

F, E, D, B, C, A

😊

S, Q, P, T, R, U

C

😊

→

R

F, C, B, A, D, E

😊

R, Q, P, T, S, U

D

😊

→

S

D, C, A, F, B, E

😊

T, U, R, Q, P, S

E

😊

→

T

C, F, E, B, A, D

😊

P, T, Q, R, U, S

F

😊

→

U

F, B, D, A, C, E

😊

完全マッチングは全部で幾つ?

男女各人数	完全マッチング数
6	720
10	3,628,800
20	2.4×10^{18}
30	2.7×10^{32}
40	8.2×10^{47}
50	3.0×10^{64}
100	9.3×10^{157}
200	#NUM!

※調べた最初の1つが安定解ならそれで計算終了だが、最悪、一番最後まで見つからないかもしれない。また、そもそも安定解など存在しないかもしれないので、その場合は全部調べなければならない

😱

完全マッチングは全部で幾つ？

完全マッチングが膨大にあるとは言っても、今のコンピュータはかなりの速さで計算できるでしょ？ だから大丈夫だね！

- 代表的なCPU, Game機, super computer の浮動小数点演算回数
 - Intel Core i7(3.2GHz) : **51.2GFLOPS** ...1秒間に約**512億**回
 - PS3 : **218GFLOPS** ...1秒間に約**2180億**回
 - PS4 : **1.84TFLOPS** ...1秒間に約**1兆8400億**回
 - 京 : **10.51PFLOPS** ...1秒間に約**1京510兆**回
- (※2011年6月, 11月 世界最速！ by Top500.org)
(※2012年6月=2位, 11月=3位, 2013年6月=4位, 11月=4位)
- ※FLOPS = **F**loating-point **O**perations **P**er **S**econd

完全マッチングを一つ見つけるのに、男(女)の人数(完全マッチング数)の浮動小数点演算できると仮定しよう

例えば、n=6(男6人, 女6人)のときは、6回の演算で計算可と仮定するということ

K(キロ)≒×10³=千倍
M(メガ)≒×10⁶=百万倍
G(ギガ)≒×10⁹=10億倍
T(テラ)≒×10¹²=1兆倍
P(ペタ)≒×10¹⁵=千兆倍
E(エクサ)≒×10¹⁸=百京倍

完全マッチングは全部で幾つ？

		51.2GFLOPS	1.84TFLOPS	10.51PFLOPS
人数	pm数	Core i7	PS4	京
6	720	0.0000001秒	0.0000000秒	0.0000000秒
10	3,628,800	0.0007088秒	0.0000197秒	0.0000000秒
20	2.4 × 10 ¹⁸	30年	306日	1.3時間
30	2.7 × 10 ³²	357,129宙齡	9,938宙齡	1.7宙齡
40	8.2 × 10 ⁴⁷	1.5E+21宙齡	4.1E+19宙齡	7.1E+15宙齡
50	3.0 × 10 ⁶⁴	6.8E+37宙齡	1.9E+36宙齡	3.3E+32宙齡
100	9.3 × 10 ¹⁵⁷	4.2E+131宙齡	1.2E+130宙齡	2.0E+126宙齡
200	#NUM!	#NUM!	#NUM!	#NUM!

圧倒的な計算力をもつコンピュータですら、**全列挙(しらみつぶし)**では答えを求めることは期待出来ない

1宙齡=138億年

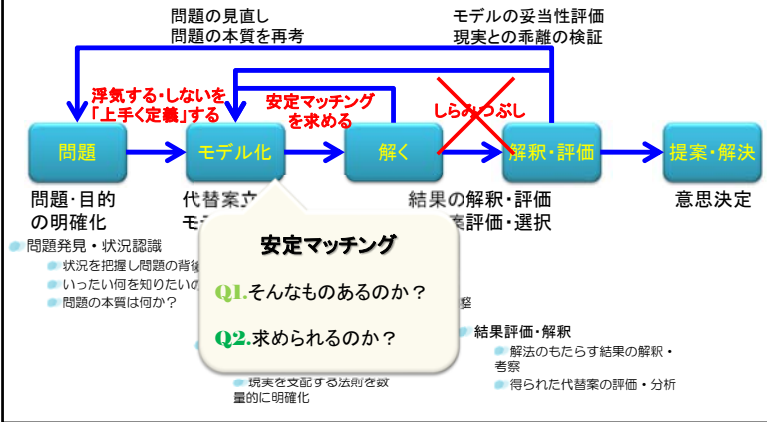


補足:スパコンの性能

- Top500** (行列**演算**:連立一次方程式を解く速度を評価)
 - 京 : **10.51PFLOPS** ...1秒間に**1京510兆**回
 - 2011年6月 1位
 - 2011年11月 1位
 - 2012年6月 2位
 - 2012年11月 3位
 - 2013年6月 4位
 - 2013年11月 4位
 - 2014年6月 4位
 - 2014年11月 4位
 - 2015年6月 4位
- 他に**Green500**なども(エネルギー消費効率の良さを競う **FLOPS per Watt**)
2015年6月上位3機は日本
1位. 富通, 2位. 青森通, 3位. 睡蓮
- ※FLOPS = **F**loating-point **O**perations **P**er **S**econd
※TEPS = **T**raversed **E**dges **P**er **S**econd
- Graph500** (大規模**グラフ解析**の性能を評価)
 - 京 : **38,621GTEPS** ...1秒間に**38兆6210億**個
 - 2014年6月 1位
 - 2014年11月 2位
 - 2015年6月 1位(約1兆個の点, 約16兆個の枝からなるグラフの幅優先探索を0.45秒で処理)
- 他にGreen500なども(エネルギー消費効率の良さを競う FLOPS per Watt)
2015年6月上位3機は日本
1位. 富通, 2位. 青森通, 3位. 睡蓮
- ✓ 計算速度
✓ アルゴリズム
✓ プログラム
などの総合力の競争

問題解決

- 「問題の把握」から「意思決定」までの流れ



ではどうする？

• 素朴で素直な方法【列挙法】

– 全ての完全マッチングをしらみつぶしに調べて、安定解を探す

時間が掛かり過ぎる！

全ての完全マッチングをしらみつぶしに調べずに、安定解を、現実的時間で見つける方法があるか？

Gale-Shapley Algorithm

人間の創造的な仕事！

安定結婚問題を解く

Gale-Shapley アルゴリズム

D.Gale & L.S.Shapley, "College admissions and the stability of marriage," American Mathematical Monthly, vol.69,p.9-15,1962.

Gale-Shapley アルゴリズム

S,Q,P,U,R,T A

R,Q,U,S,P,T B

S,Q,P,T,R,U C

R,Q,P,T,S,U D

T,U,R,Q,P,S E

P,T,Q,R,U,S F

結婚して~

結婚して~

結婚して~

OK

OK

P E,A,F,D,C,B

Q F,E,D,B,C,A

R F,C,B,A,D,E

S D,C,A,F,B,E

T C,F,E,B,A,D

U F,B,D,A,C,E

Gale-Shapley アルゴリズム

S,Q,P,U,R,T A

R,Q,U,S,P,T B

S,Q,P,T,R,U C

R,Q,P,T,S,U D

T,U,R,Q,P,S E

P,T,Q,R,U,S F

そんなー

結婚して~

Bye!

OK

P E,A,F,D,C,B

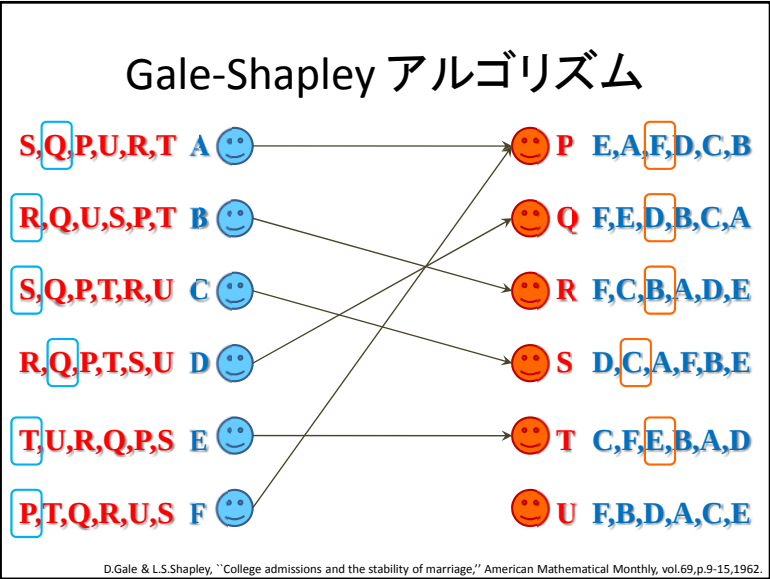
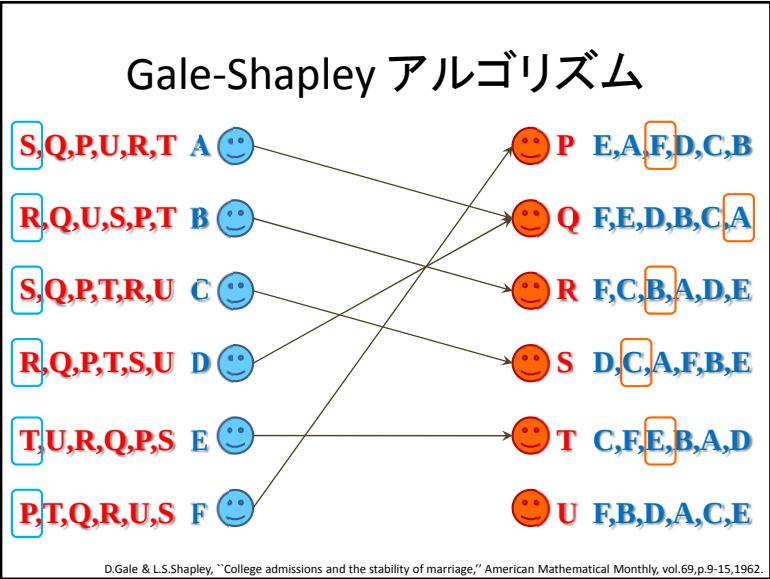
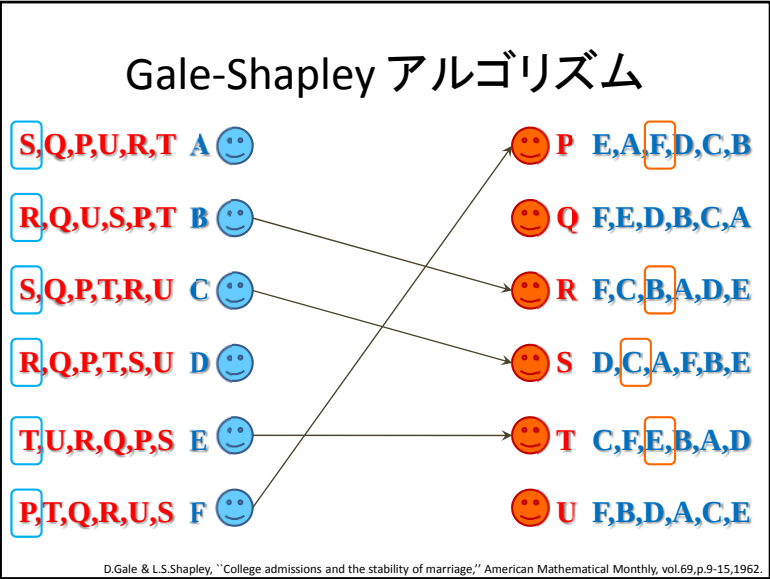
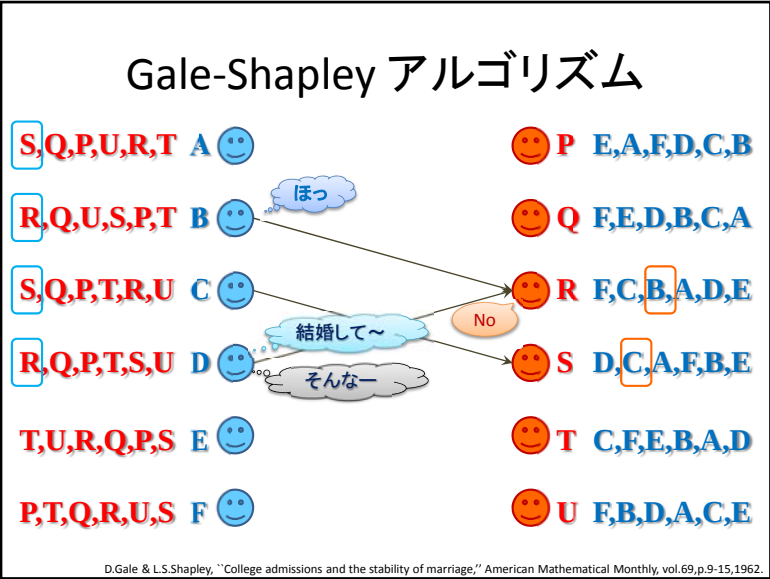
Q F,E,D,B,C,A

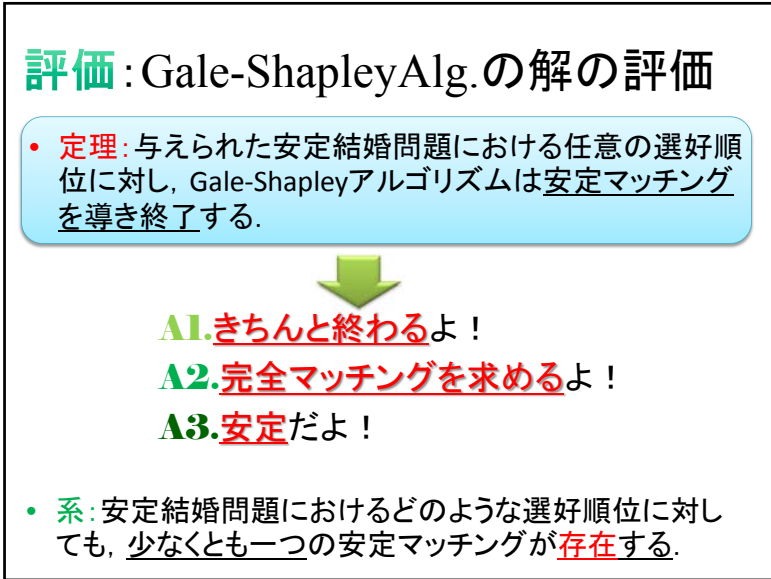
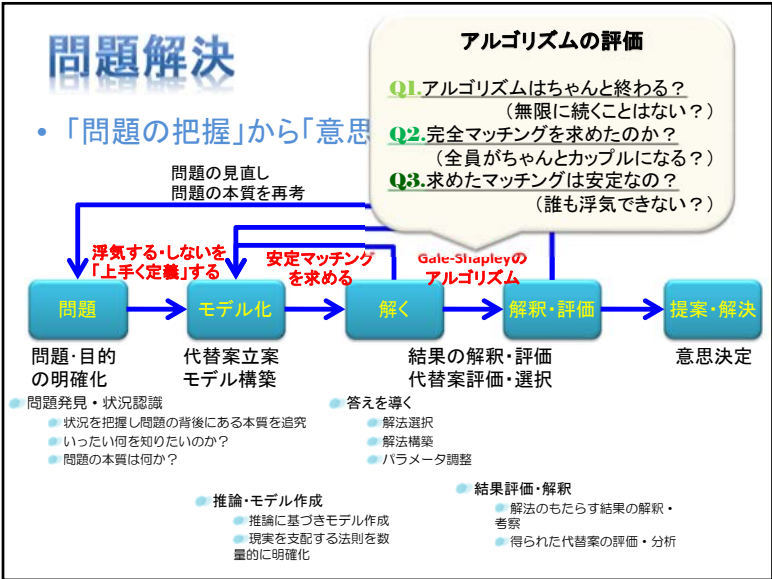
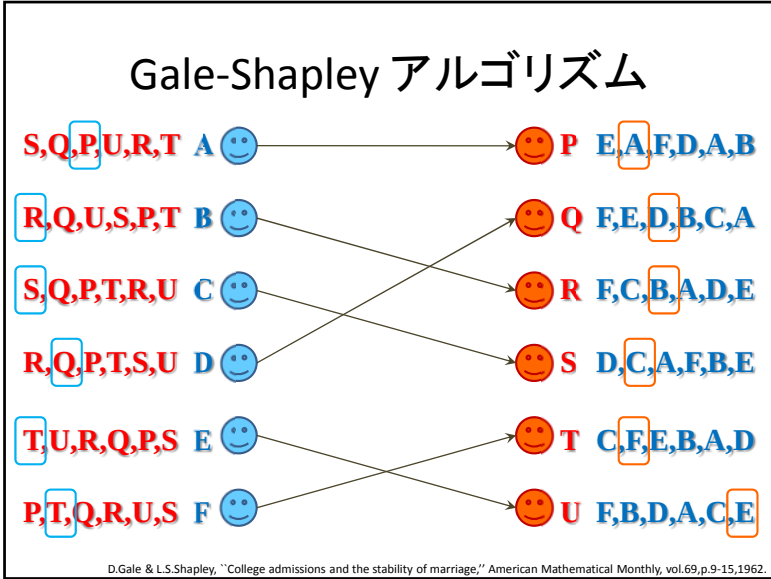
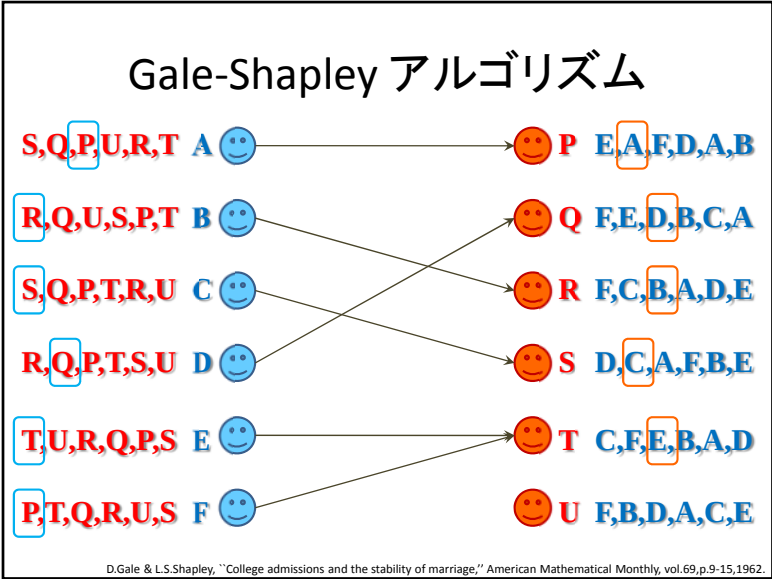
R F,C,B,A,D,E

S D,C,A,F,B,E

T C,F,E,B,A,D

U F,B,D,A,C,E





評価: Gale-Shapley Alg. って速いの？

• 男(女)の数を n とすると, 大雑把な見積もりで,

$O(n^2)$ 多項式オーダー

コンピュータに計算させてみよう！

簡単のため $10n^2$ の浮動小数点演算回数で計算できると仮定

人数	pm数	京 & しらみつぶし	Core i7 & GS Alg
6	720	0.0000000 秒	0.0000000 秒
10	3,628,800	0.0000000 秒	0.0000000 秒
20	2.4×10^{18}	1.3 時間	0.0000001 秒
30	2.7×10^{32}	1.7 宙齡	0.0000002 秒
40	8.2×10^{47}	7.1E+15 宙齡	0.0000003 秒
50	3.0×10^{64}	3.3E+32 宙齡	0.0000005 秒
100	9.3×10^{157}	2.0E+126 宙齡	0.0000020 秒
200	#NUM!	#NUM!	0.0000078 秒
1000	#NUM!	#NUM!	0.0001953 秒
10000	#NUM!	#NUM!	0.0195313 秒
100000	#NUM!	#NUM!	1.9531250 秒
1000000	#NUM!	#NUM!	195.3125000 秒

世界最速SuperComp
+ 力技 (しょぼい方法)

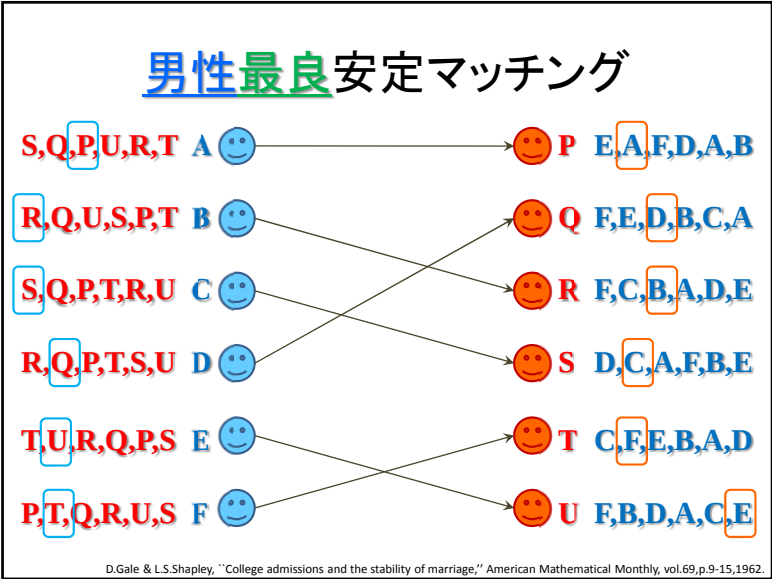
<<<

そこらのPC
+ 人間の知恵

評価: Gale-Shapley Alg. の解の評価2

• 定理: 男性側のプロポーズの順番に関係なく, Gale-Shapleyアルゴリズムは, 同一の安定マッチングを導く.

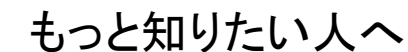
• 系: 安定結婚問題におけるどのような選好順位に対しても, Gale-Shapleyアルゴリズムは, 男性側からプロポーズすれば男性最良安定マッチングを導く.





- 定理:** 与えられた安定結婚問題について,
男性最良安定マッチング = 女性最悪安定マッチング
男性最悪安定マッチング = 女性最良安定マッチング
 である.

➡ 教訓!?『待ってちゃダメ！
好きになったら自分から告白しなさい』



- 関連する経営学科の授業
 - －「ネットワークモデル分析」(4セメ)
 - －「最適化モデル分析」(5セメ)
 - －「意思決定科学」(6セメ)

etc...

練習:

男性最良安定マッチングを求めよ(プロポーズは上から順に一人ずつ)

1234

PQRSA

1234

PQAB

1234

PSQR

1234

QSRP

1234

RQP

1234

PABCD

1234

QACBD

1234

RADCB

1234

SBCDA

Gale-Shapley Algorithm
のプロポーズ順

1

→

2

→

3

→

4

→

5

→

6

→

7

→

8

→

9

→

10

→

11

→

12

→

13

→

14

→

練習:解答例

男性最良安定マッチングを求めよ(プロポーズは上から順に一人ずつ)

1234

PQRSA

1234

PQAB

1234

PSQR

1234

QSRP

1234

RQP

1234

PABCD

1234

QACBD

1234

RADCB

1234

SBCDA

Gale-Shapley Algorithm
のプロポーズ順

1

A → P

2

B → P

3

→

4

→

5

→

6

→

7

→

8

→

9

→

10

→

11

→

12

→

13

→

14

→

練習:解答例

男性最良安定マッチングを求めよ(プロポーズは上から順に一人ずつ)

1234

PQRSA

1234

PQAB

1234

PSQR

1234

QSRP

1234

RQP

1234

PABCD

1234

QACBD

1234

RADCB

1234

SBCDA

Gale-Shapley Algorithm
のプロポーズ順

1

A → P

2

B → P

3

C → Q

4

D → R

5

→

6

→

7

→

8

→

9

→

10

→

11

→

12

→

13

→

14

→

練習:解答例

男性最良安定マッチングを求めよ(プロポーズは上から順に一人ずつ)

1234

PQRSA

1234

PQAB

1234

PSQR

1234

QSRP

1234

RQP

1234

PABCD

1234

QACBD

1234

RADCB

1234

SBCDA

Gale-Shapley Algorithm
のプロポーズ順

1

A → P

2

B → P

3

C → Q

4

D → R

5

A → Q

6

C → S

7

→

8

→

9

→

10

→

11

→

12

→

13

→

14

→