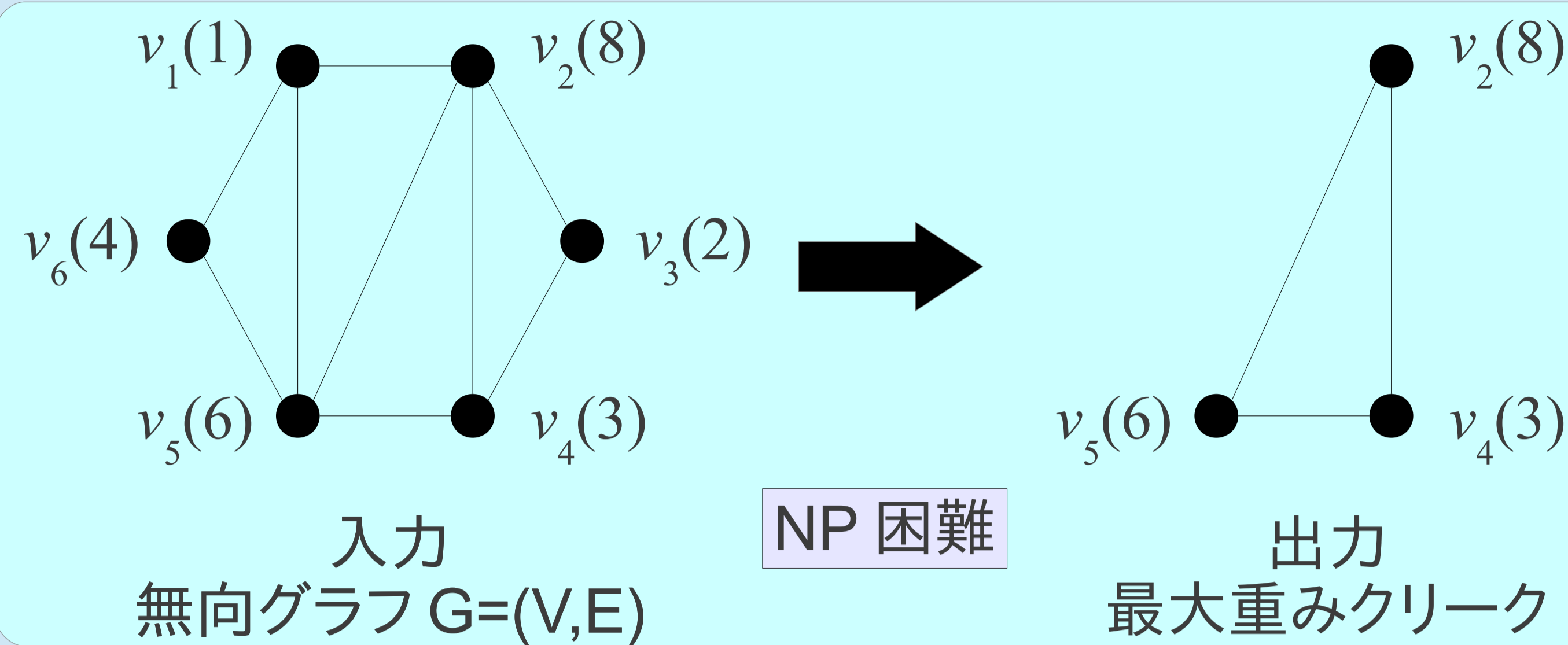


最大重みクリークに対する最適解テーブル法

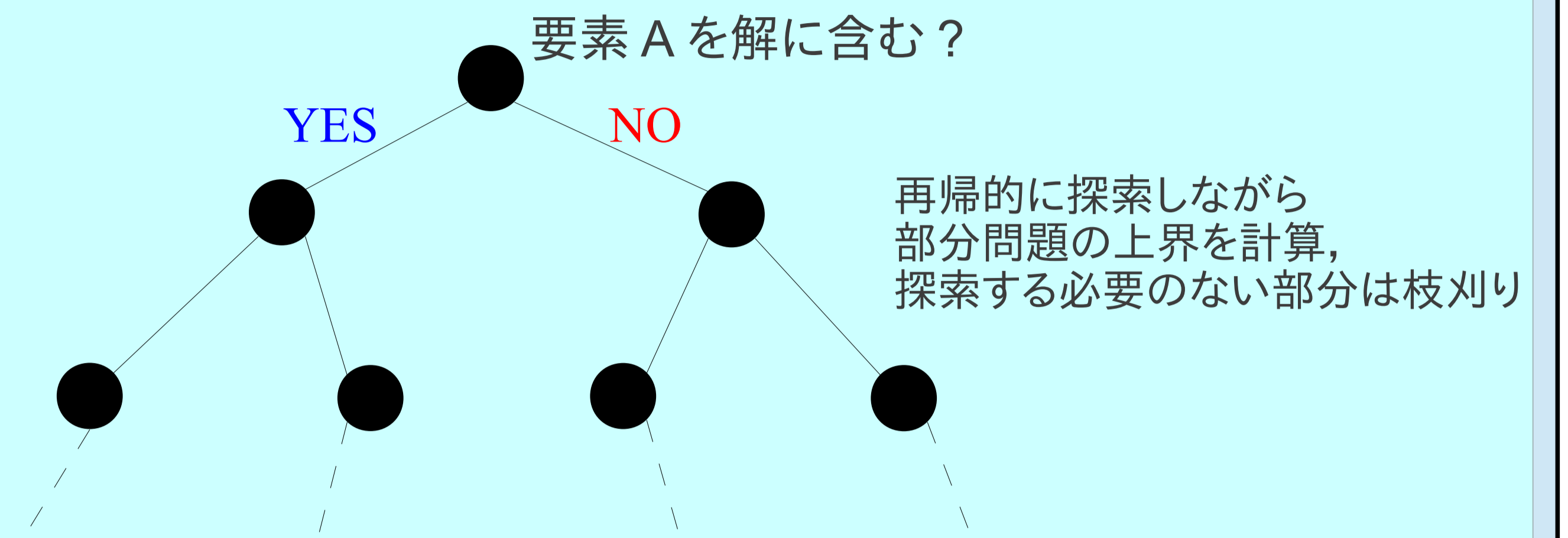
神戸大学大学院工学研究科

©清水悟司 山口一章 斎藤寿樹 増田澄男

最大重みクリーク問題



分枝限定法による厳密解法

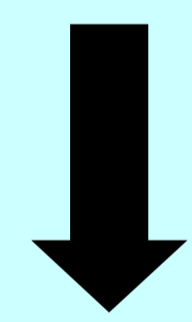


提案法

● 提案法の流れ

前処理

1. 頂点集合の分割
2. 最適解テーブルの作成



分枝限定法

最適解テーブルを利用して上界計算を高速に行う

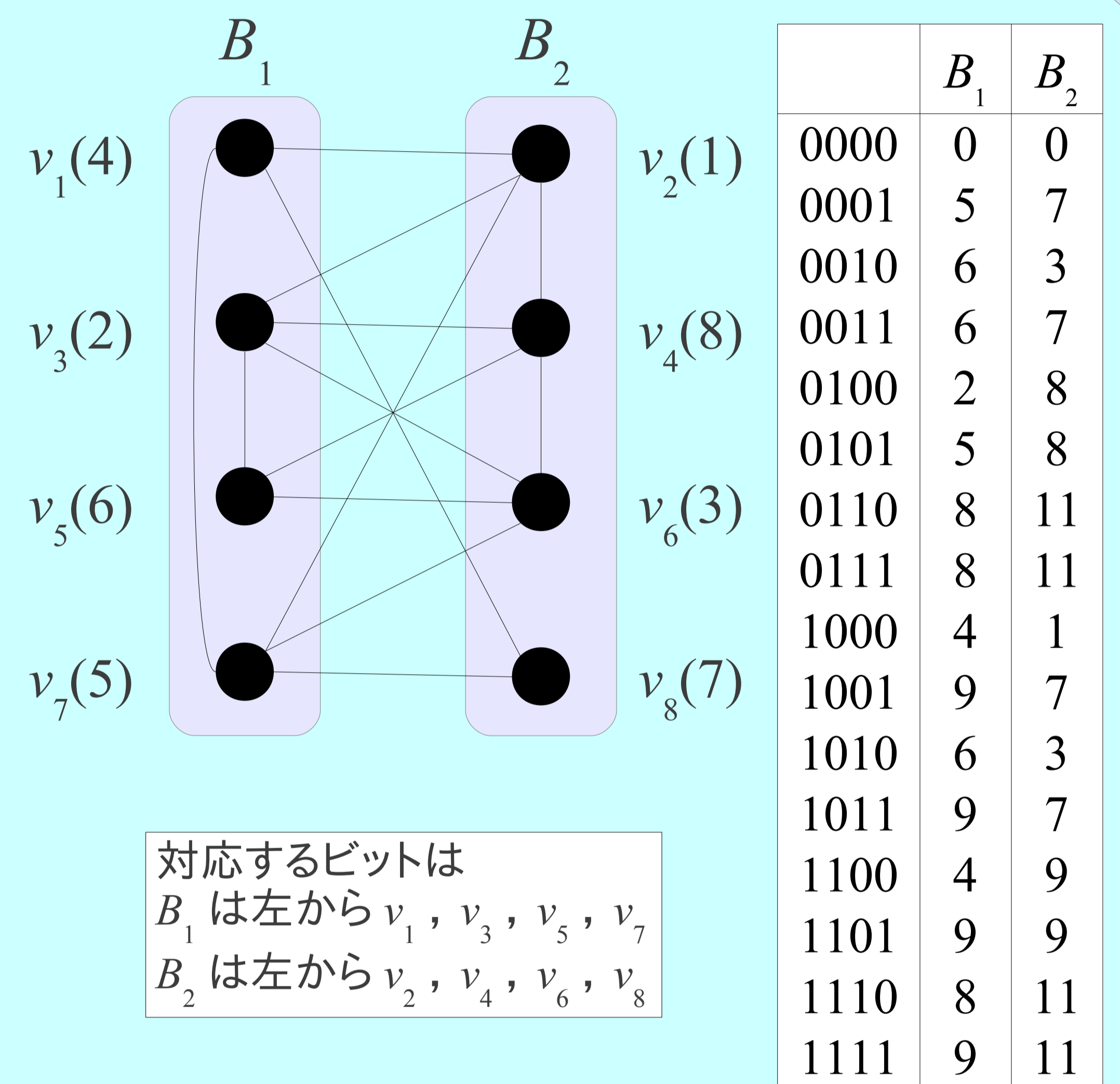
● 前処理

1. 頂点集合の分割

頂点集合を B_1, B_2, \dots, B_k に分割.

2. 最適解テーブルの作成

各 B_i 全ての部分集合の最適解の重みを求める。後で取り出せるようにテーブルに保存する。

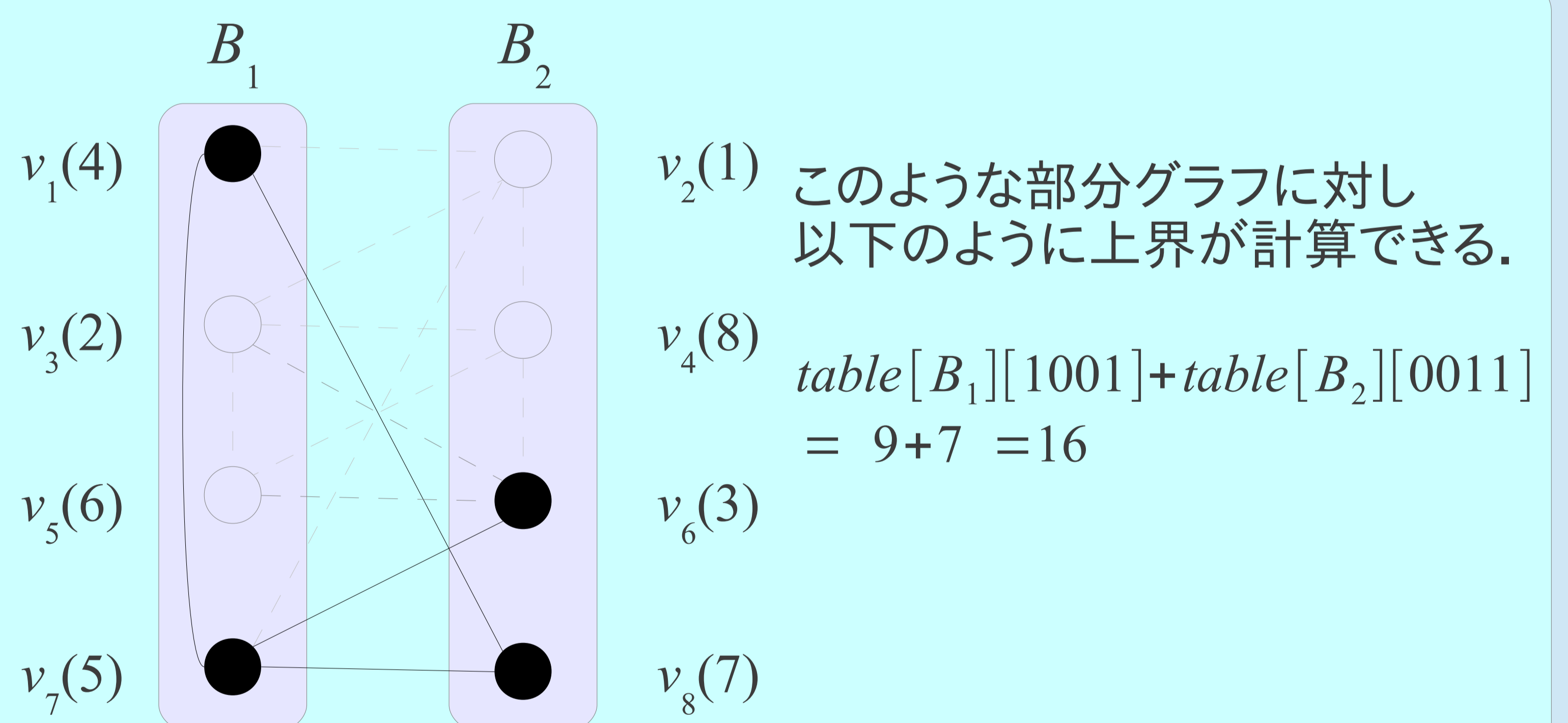


● 上界計算

$$\text{最適解の重み} \leq \sum_{i=1}^k (\text{各 } B_i \text{ の最適解の重み})$$

この関係は部分グラフに対しても成り立つ。

最適解テーブルを利用すれば
配列の参照だけで上界が計算できる。
上界計算にあまり時間がかからない。



計算機実験

以下のベンチマークで有効性を確認。

- ランダムグラフ (重みは 1 から 10 の整数)
- 組合せオークション
- error correcting code
- DIMACS (重みなし)

ランダムグラフ以外でも提案法が高速であった。また、従来法と違い、苦手なインスタンスがないことを確認。

OS : Linux
 言語 : C / C++ (gcc 4.4.6)
 CPU : i7-2600 3.40GHz
 メモリ : 8GB
 ブロックの要素数は 20 から 25

ランダムグラフの計算時間(s)						
頂点数	辺密度	提案法	cliquer	YM	vctable	DK
4000	0.1	0.68	0.36	1.15	0.86	0.94
3000	0.2	2.06	1.36	6.08	2.70	5.99
2500	0.3	14.39	10.03	37.84	15.95	44.53
1500	0.4	13.05	15.06	42.29	14.45	58.04
1000	0.5	12.63	28.67	61.50	17.85	94.84
700	0.6	17.44	64.38	99.07	29.99	212.70
500	0.7	33.55	212.79	201.98	66.35	674.06
300	0.8	17.90	242.38	93.70	52.88	511.97
200	0.9	11.71	>1000	89.85	60.44	409.55
200	0.95	179.58	>1000	>1000	909.88	>1000
200	0.98	14.80	>1000	>1000	40.44	967.34