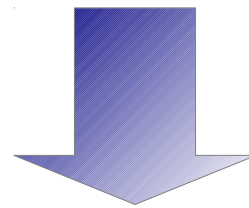
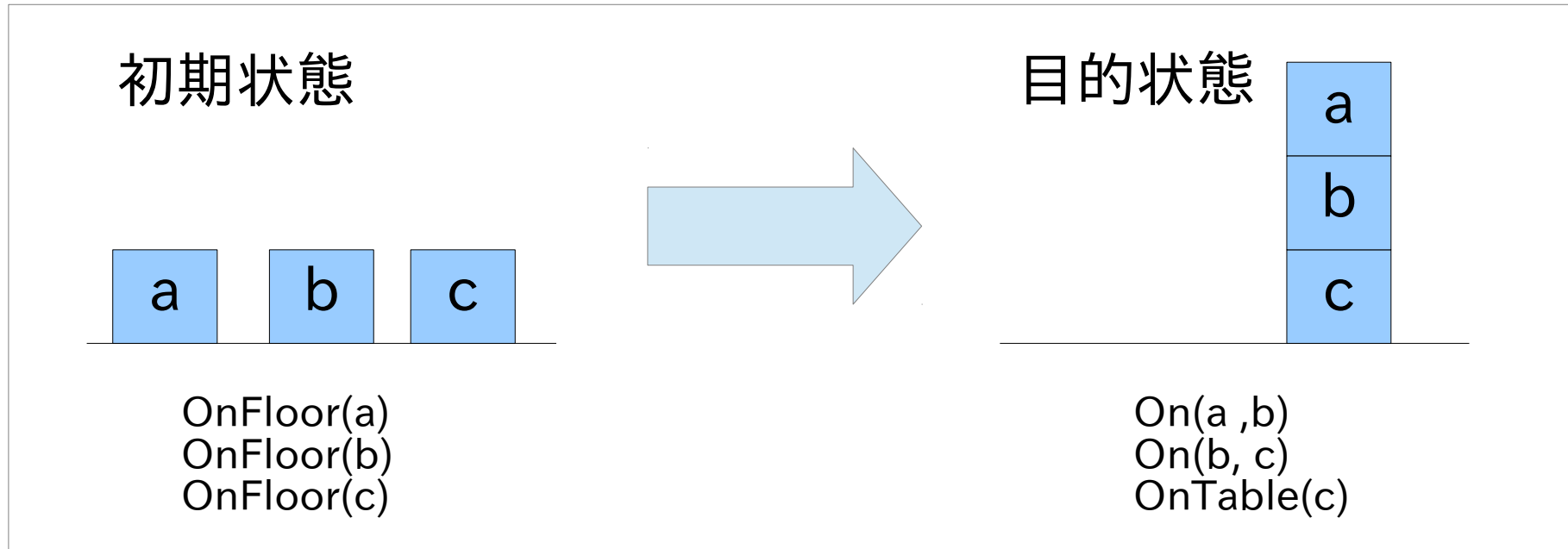


セル生産システムに対する 汎用プランニング技術の適用について

広域科学科 4年
08-120401
遠藤翔馬

汎用プランニング

入力: 初期状態と目的状態



どういう行動を取れば初期状態から
目的状態に移行できるか？

出力: プラン (=行動の順序列)

「cの上にbを置く」→「bの上にaを置く」

汎用プランニングの実問題への応用： 工場での流れ作業

- ライン生産方式とセル生産方式



ライン生産方式

- 少品種大量生産
- 多人数で1工程

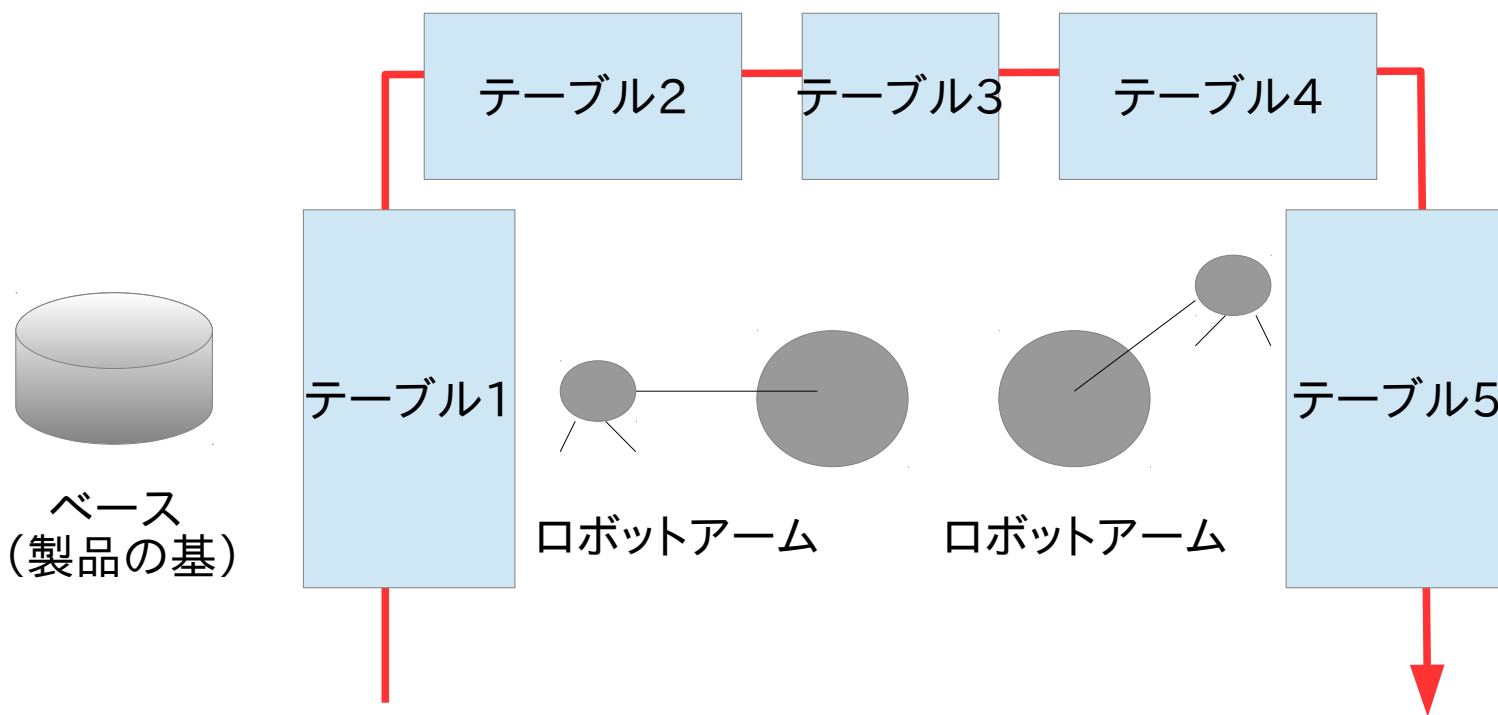


セル生産方式

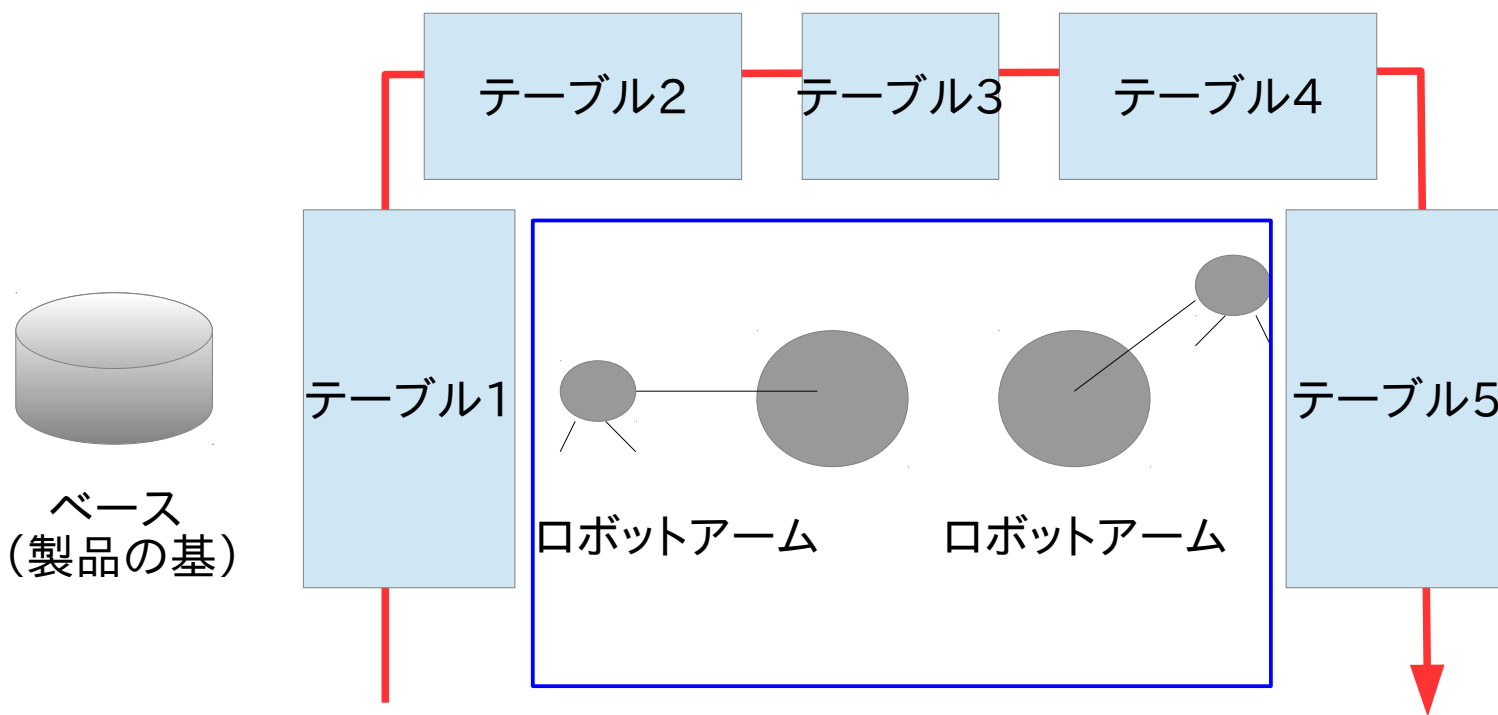
- 多品種少量生産
- 1人で1工程
- 作る製品の切り替えが容易

柔軟性が高いぶん、組立手順
(=プラン)を立案することが難しい

セル生産システムのモデル化例

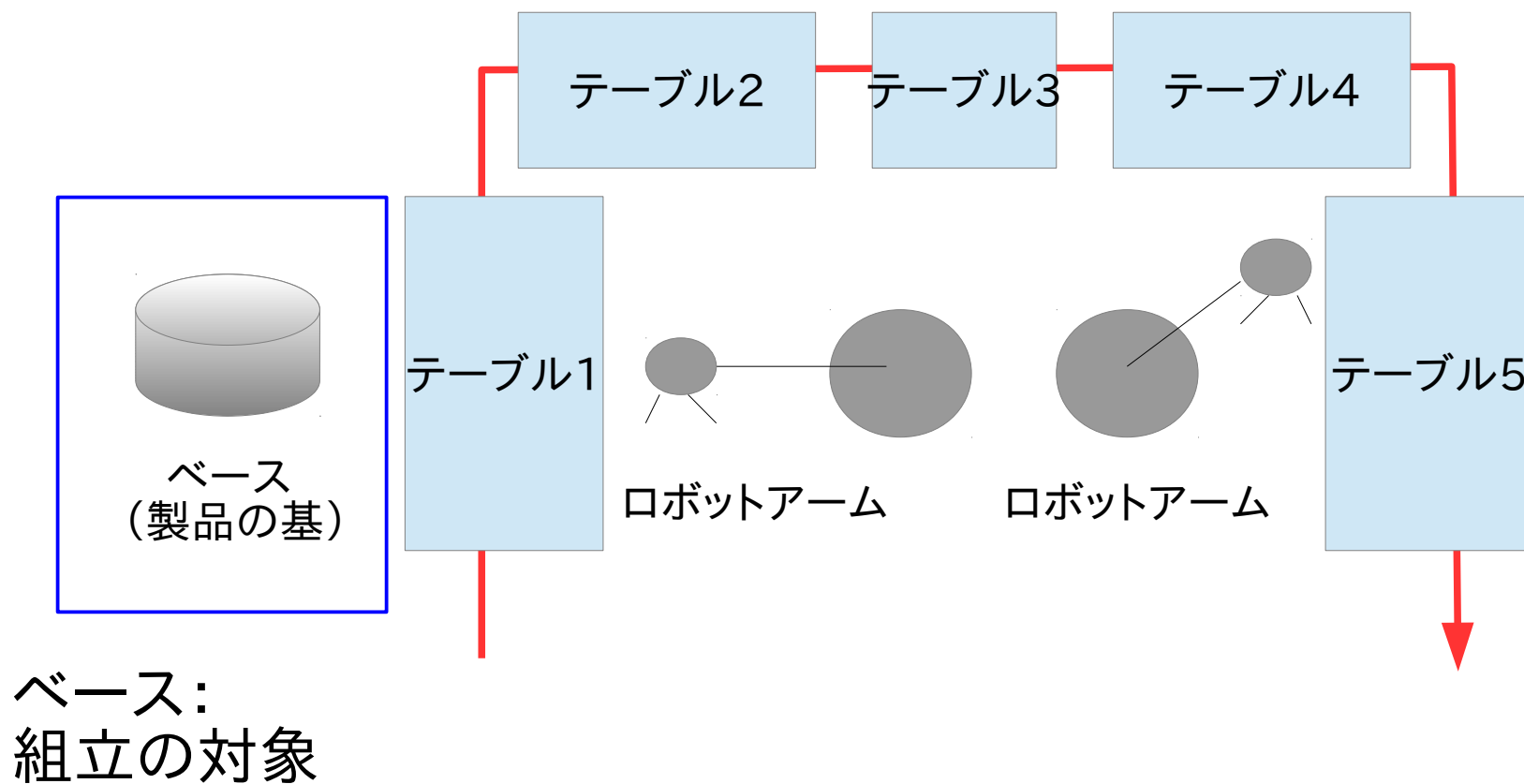


セル生産システムのモデル化例

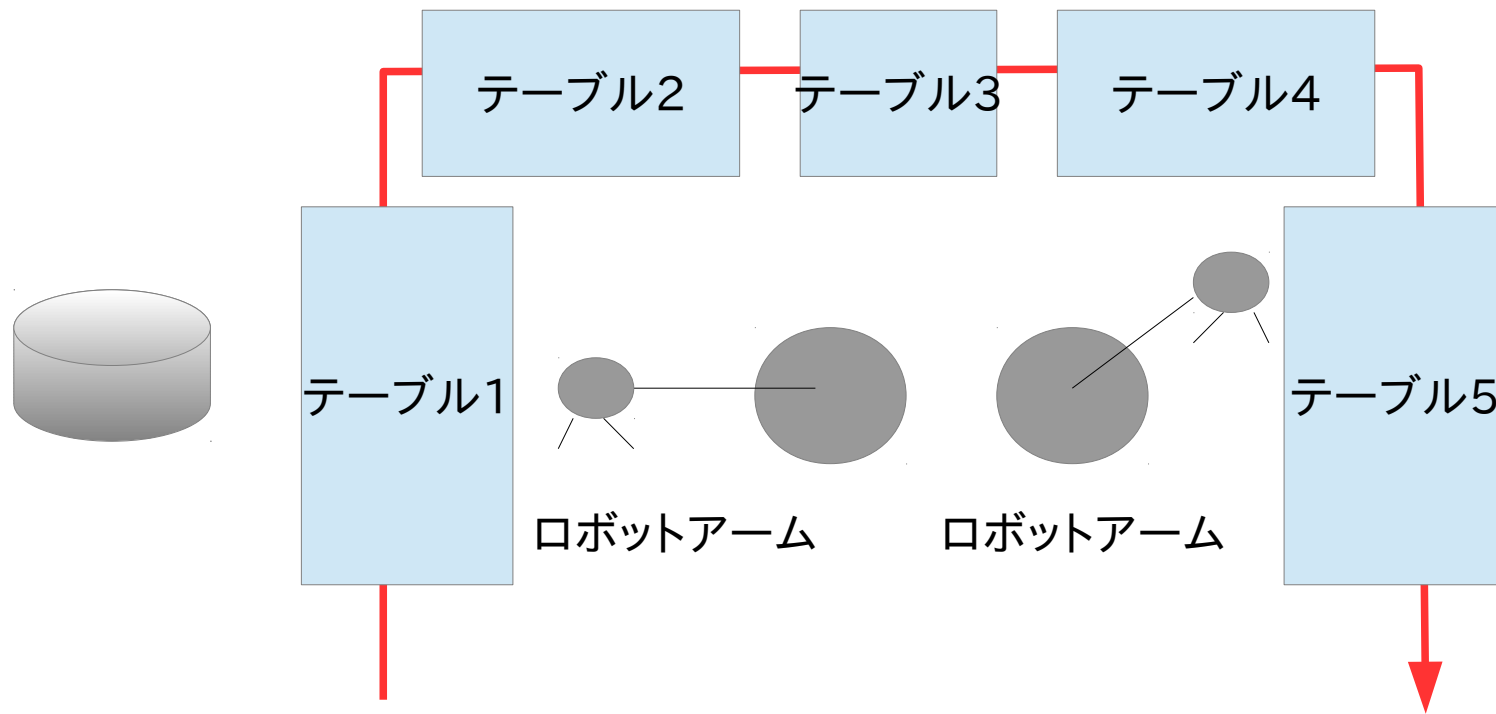


ロボットアーム:
物体を掴んで組立作業
を行う

セル生産システムのモデル化例

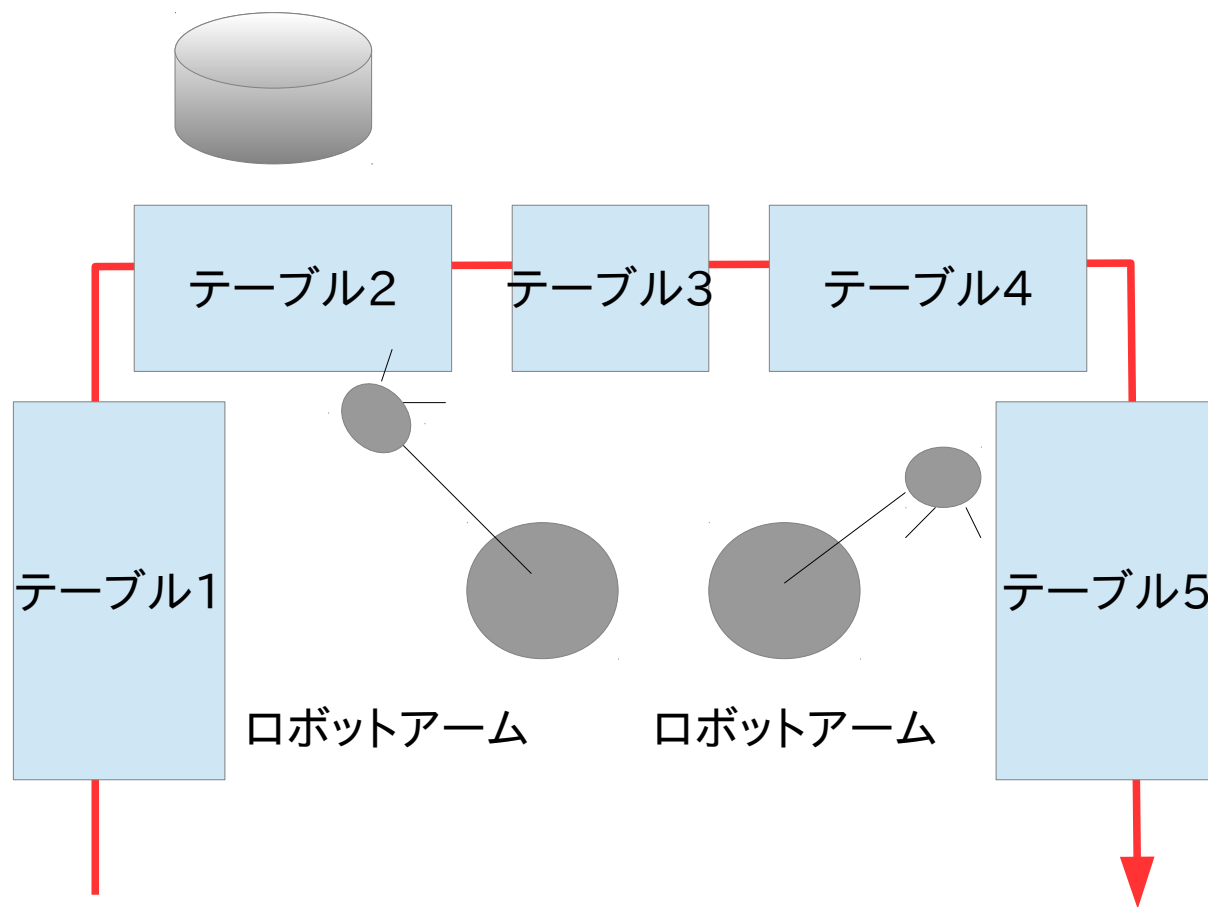


セル生産システムのモデル化例



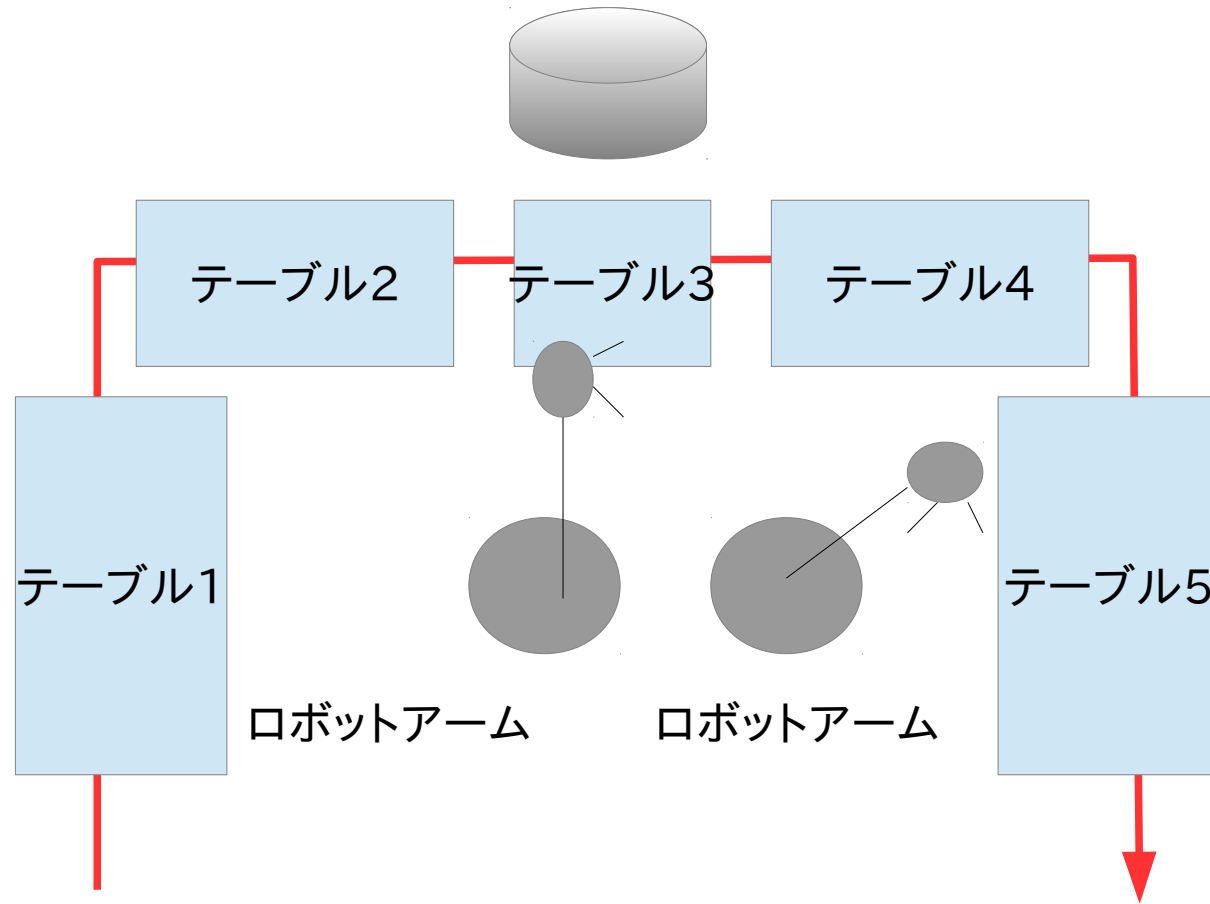
決まった順序で組立作業が進む
ただしアームの動かし方に制約はない

セル生産システムのモデル化例



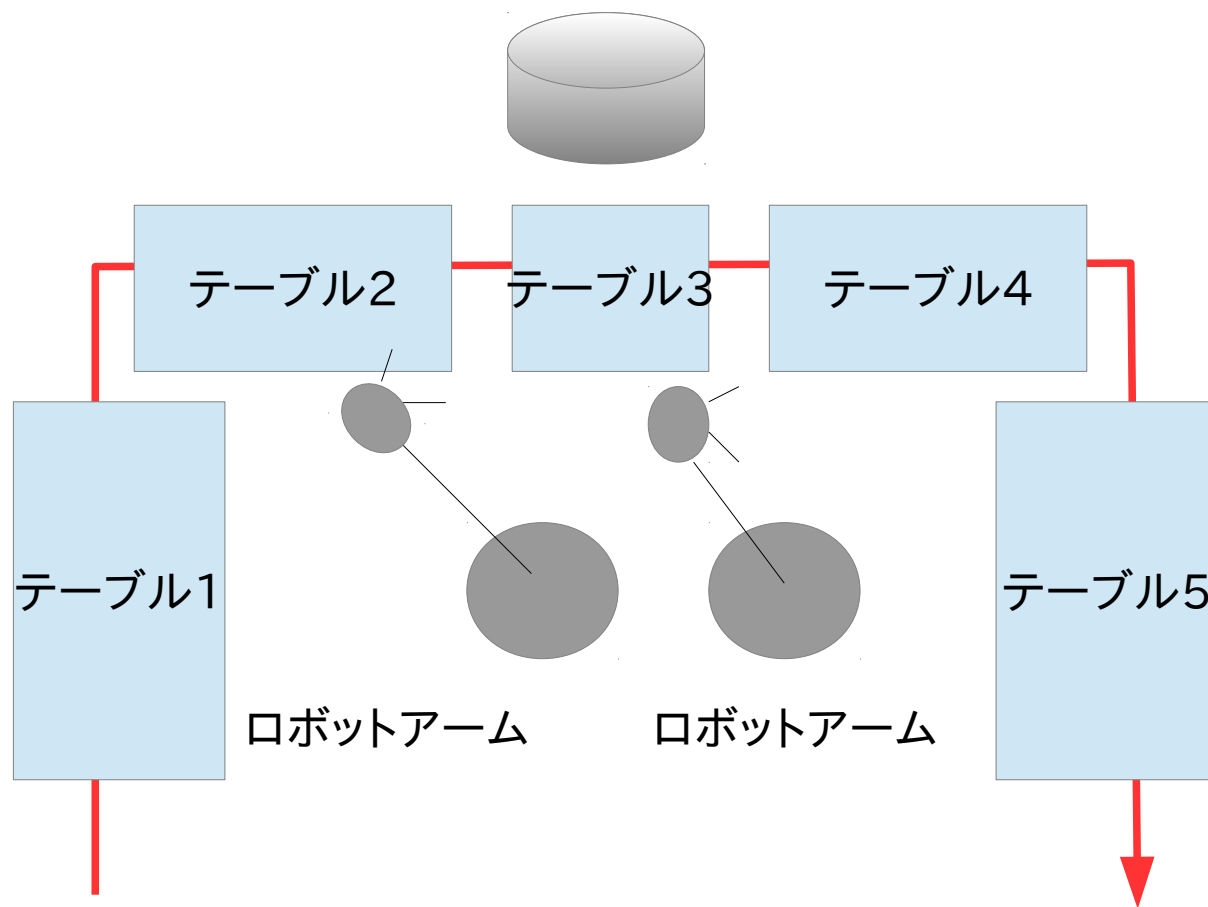
決まった順序で組立作業が進む
ただしアームの動かし方に制約はない

セル生産システムのモデル化例



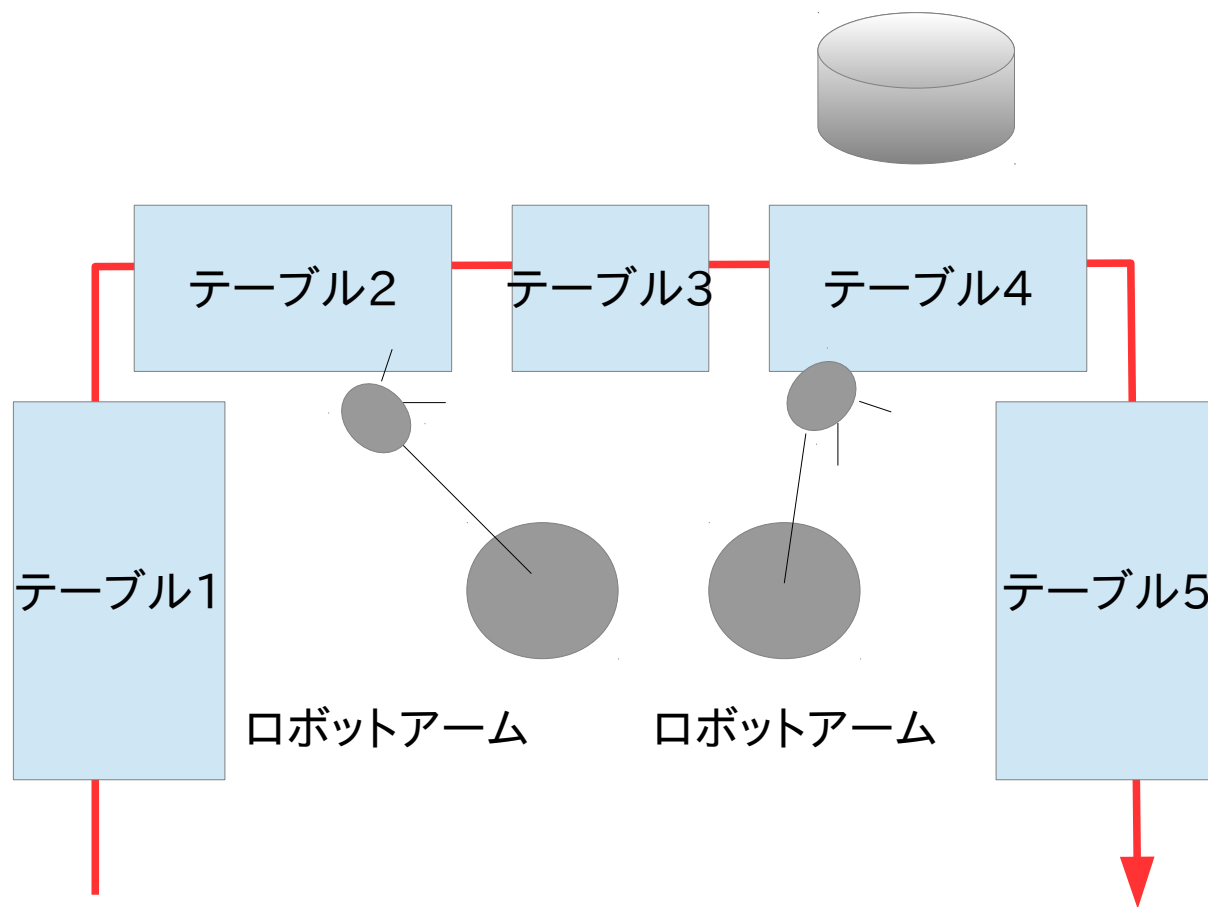
決まった順序で組立作業が進む
ただしアームの動かし方に制約はない

セル生産システムのモデル化例



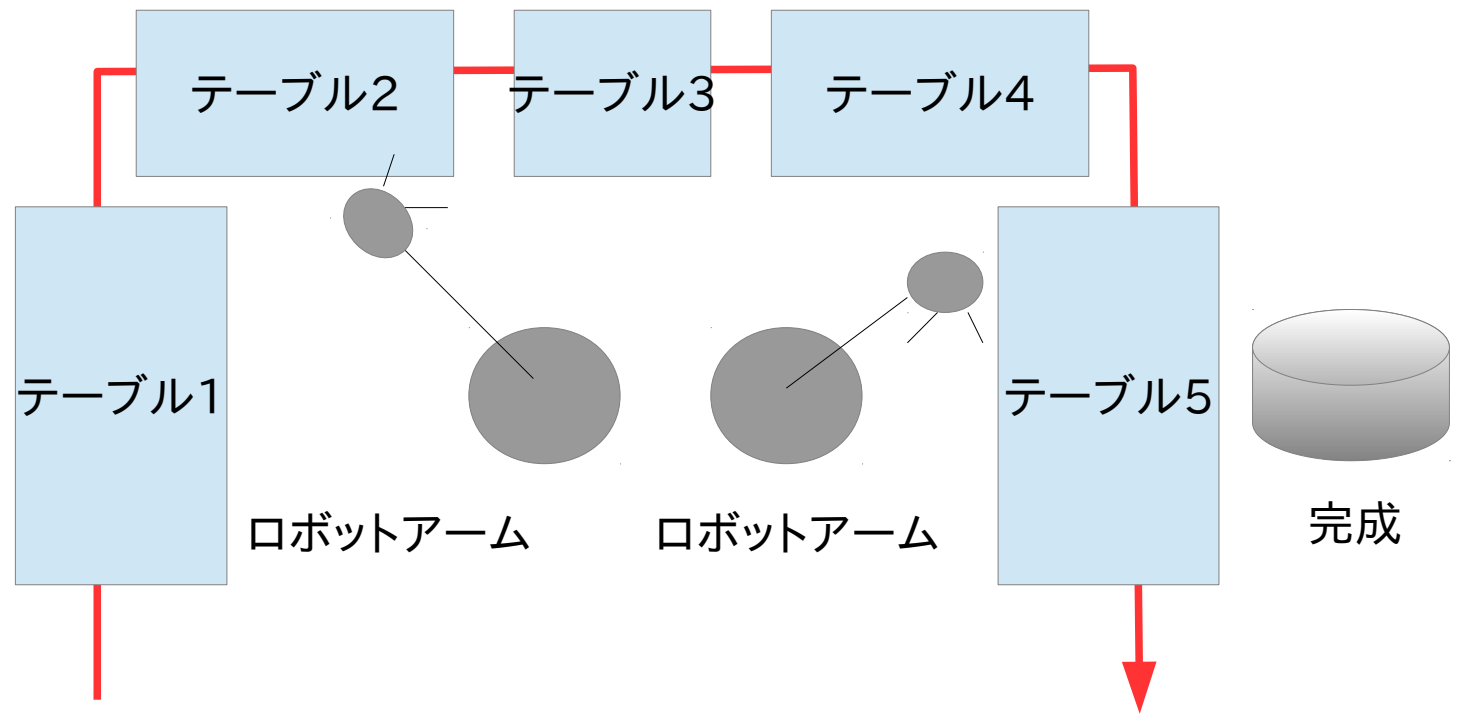
決まった順序で組立作業が進む
ただしアームの動かし方に制約はない

セル生産システムのモデル化例



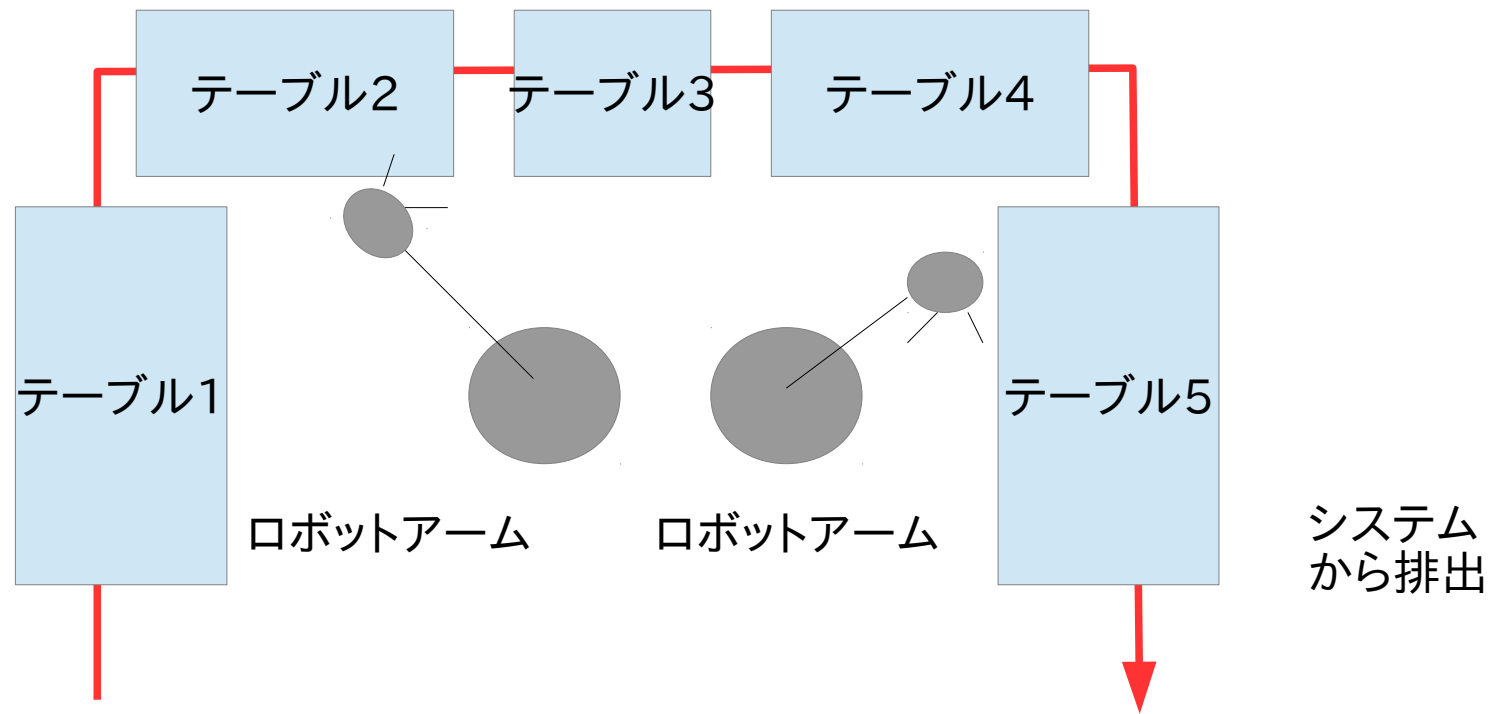
決まった順序で組立作業が進む
ただしアームの動かし方に制約はない

セル生産システムのモデル化例



決まった順序で組立作業が進む
ただしアームの動かし方に制約はない

セル生産システムのモデル化例



決まった順序で組立作業が進む
ただしアームの動かし方に制約はない

先行研究(Ochi 2013)の実験

- 汎用プランナを用いたセル生産システムモデルのプランニング
- セル生産方式に従来の単純なプランニングを適用しても上手くいかないことを示した実験
- 最適解(もっとも効率良い手順)を求める設定では、製品を3つ作らせようとしただけでパンクした
 - 他の設定はパンクしなかったものの、熟練者が手作業で組んだプランより2倍前後効率が悪い
 - 効率: 生成したプランのコスト(N個の製品を組み立てるための時間)

問題と改善案

引き続き先行研究(Ochi 2013)より

- 汎用プランニングがうまくいかない原因
→探索空間の組み合わせ爆発
- 実験で用いたプランニングモデルは、流れ作業が
大部分は同じ作業の繰り返しになるという特性を
全く考慮に入れていない
→ループに相当する概念をモデルに導入する

イメージ) ベタ書きからループへ

```
move(A0)  
paint(A0)  
attach(B0)
```

```
move(A1)  
paint(A1)  
attach(B1)
```

```
move(A2)  
...
```



```
for ( i=0; i<10; i++)  
  move(Ai)  
  paint(Ai)  
  attach(Bi)
```

steady-stateモデルの導入

引き続き先行研究(Ochi 2013)より

- steady-stateモデル

ループの概念を表現可能にするプランニングモデル

- 形式上は一般的なプランニングモデルと同じ

入力: 初期状態と目的状態

出力: 初期状態→目的状態に至るプラン

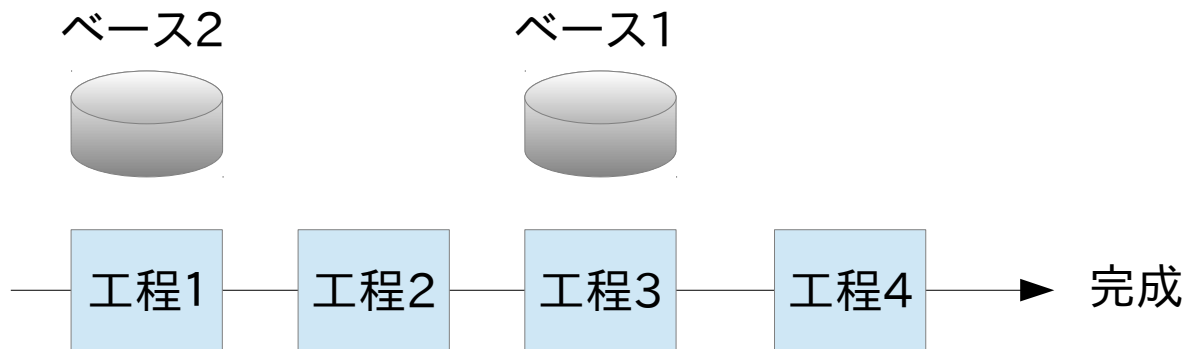
- 何が違うのか?

初期状態と目的状態が**ほぼ同じ**形をとる

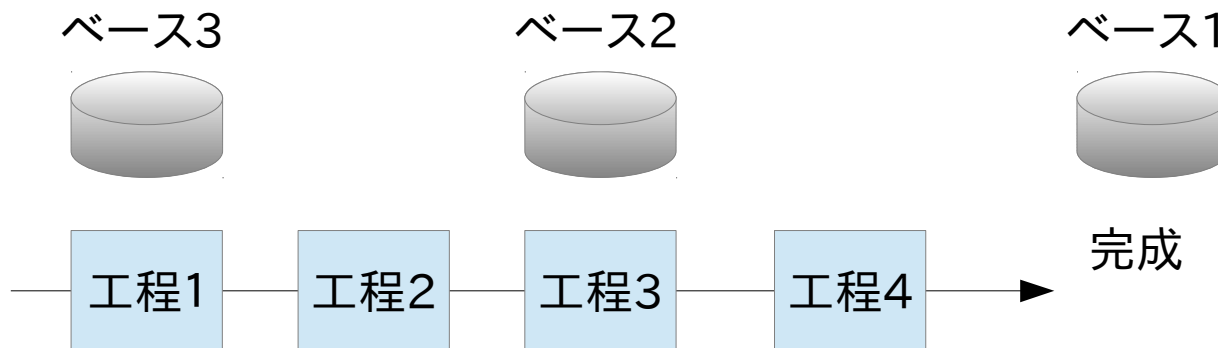
これを解いて得られるプランが**ループの1周分**に対応

例：steady-stateモデルにおける 初期状態と目的状態

初期状態

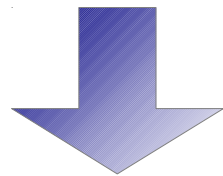
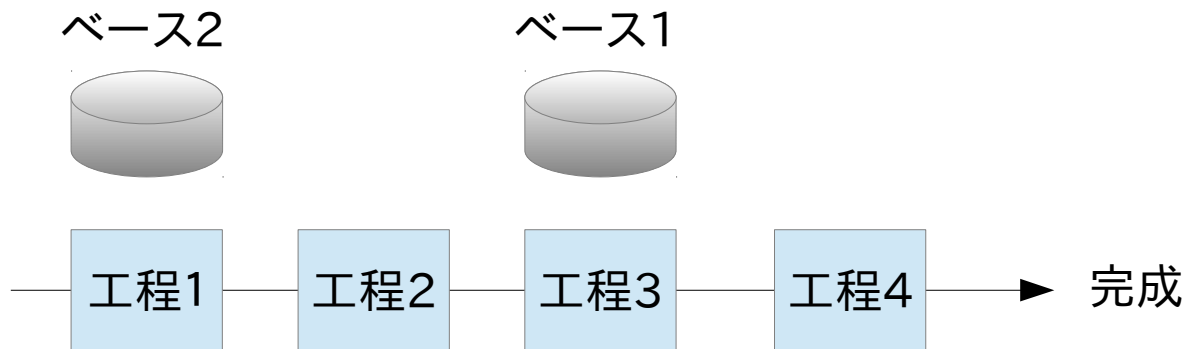


目的状態



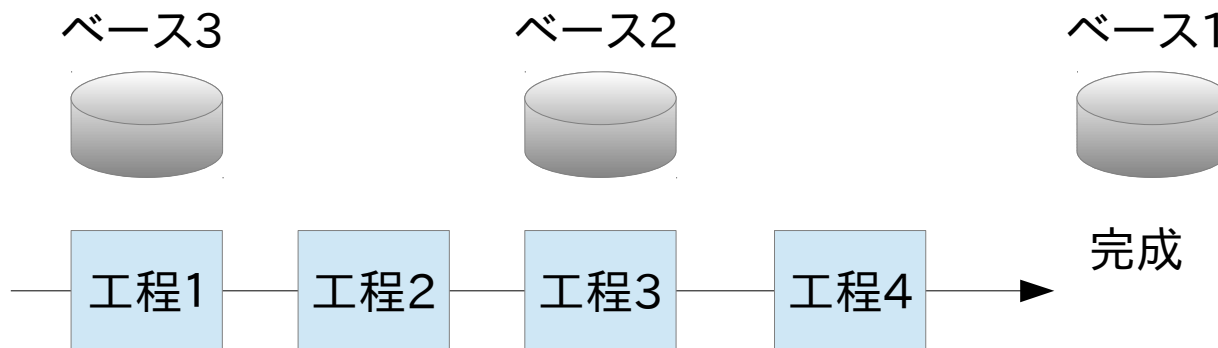
steady-stateモデルのループ表現

初期状態



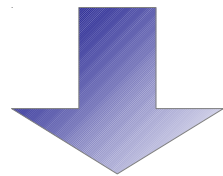
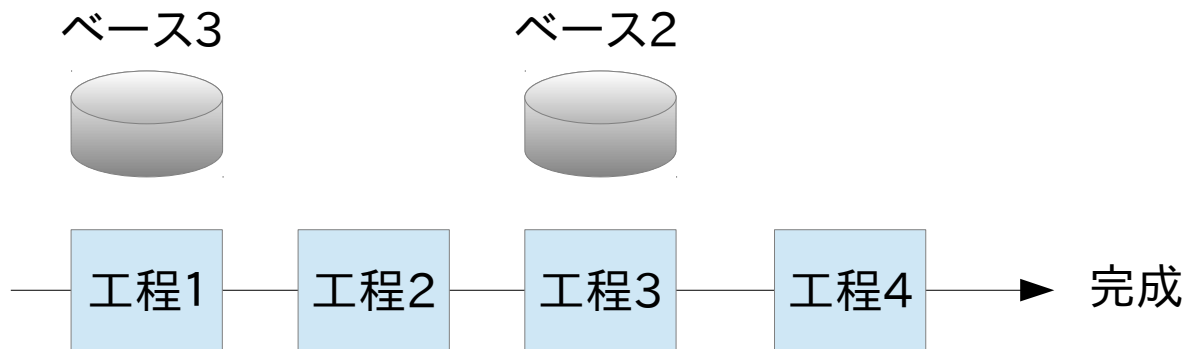
一度この短いプランを求めてしまえば、
あとはベースの番号を書き換えるだけで
大量生産に対応可能

目的状態



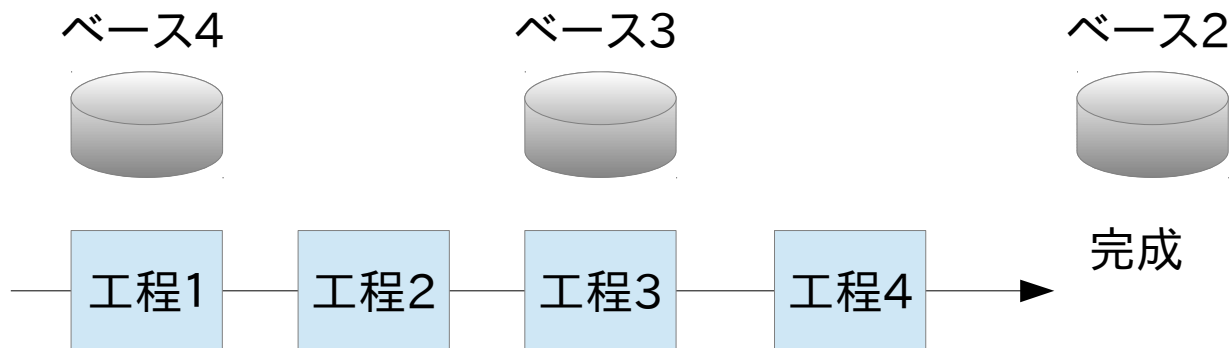
steady-stateモデルのループ表現

初期状態



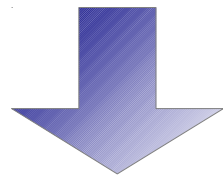
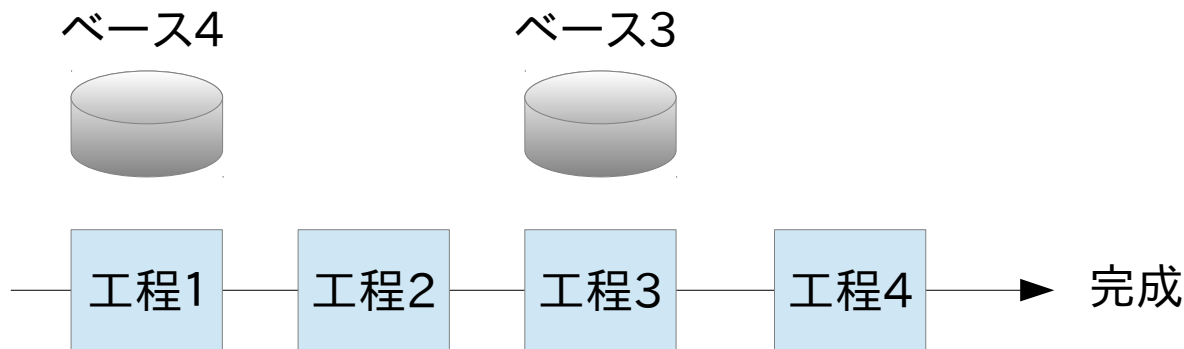
一度この短いプランを求めてしまえば、
あとはベースの番号を書き換えるだけで
大量生産に対応可能

目的状態



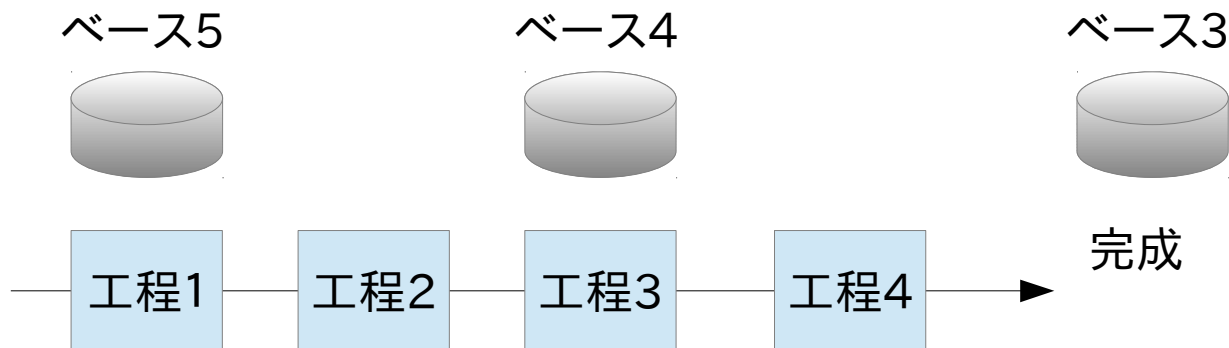
steady-stateモデルのループ表現

初期状態



一度この短いプランを求めてしまえば、
あとはベースの番号を書き換えるだけで
大量生産に対応可能

目的状態



steady-stateモデルの効果

- 立てるべきプランの長さを削減した
- プランニング問題の複雑さは立てるプランの長さに応じて指数的なオーダーで増える
 - 長さの削減は複雑さの軽減に大きく有効

Ochi(2013)における steady-stateモデルの課題

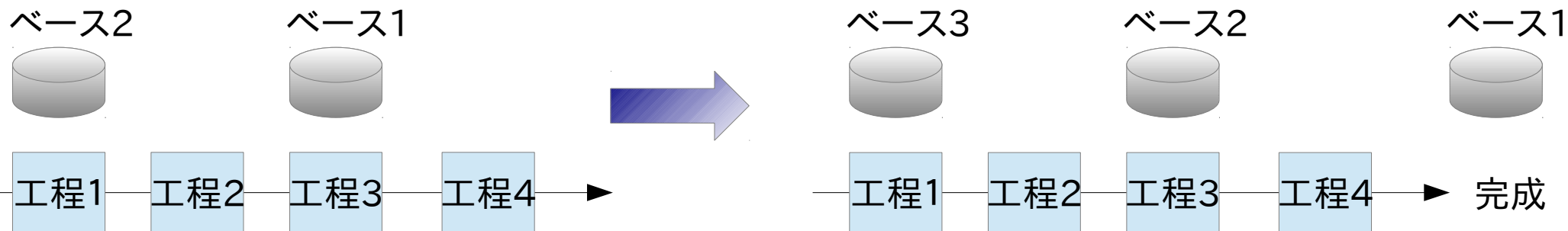
- steady-stateモデル化のための変換を手作業で行なっている
- 「ループ点」の決め方には複数の選択肢がある

例えば...

Ochi(2013)における steady-stateモデルの課題

- steady-stateモデル化のための変換を手作業で行なっている
- 「ループ点」の決め方には複数の選択肢がある

例えば...

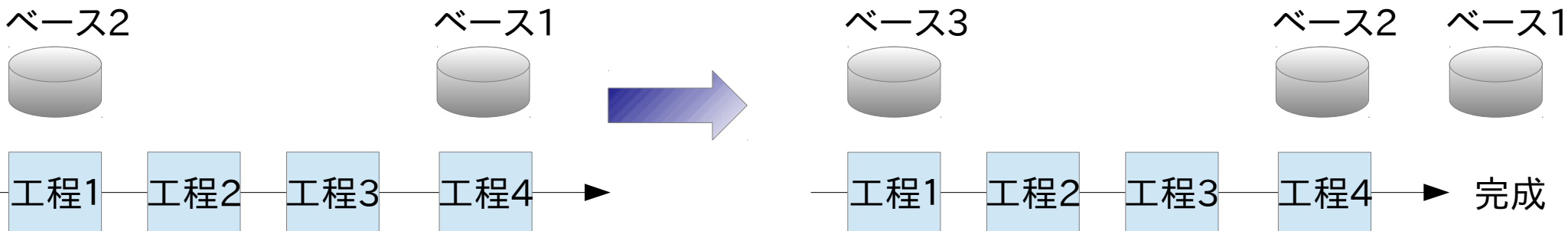


という形でループを回してもよいが

Ochi(2013)における steady-stateモデルの課題

- steady-stateモデル化のための変換を手作業で行なっている
- 「ループ点」の決め方には複数の選択肢がある

例えば...

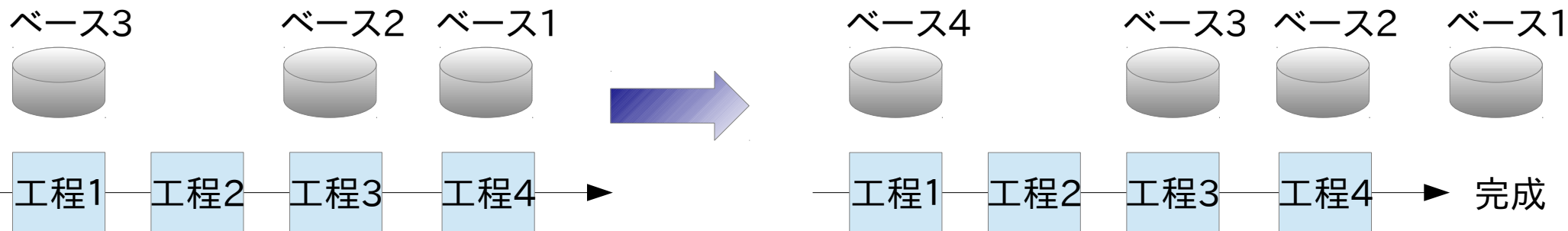


こういうループも考えられる

Ochi(2013)における steady-stateモデルの課題

- steady-stateモデル化のための変換を手作業で行なっている
- 「ループ点」の決め方には複数の選択肢がある

例えば...



あるいはこういうものも

Ochi(2013)における steady-stateモデルの課題

- steady-stateモデル化のための変換を手作業で行なっている
- 「ループ点」の決め方には複数の選択肢がある

→どう選んだらいいのか？

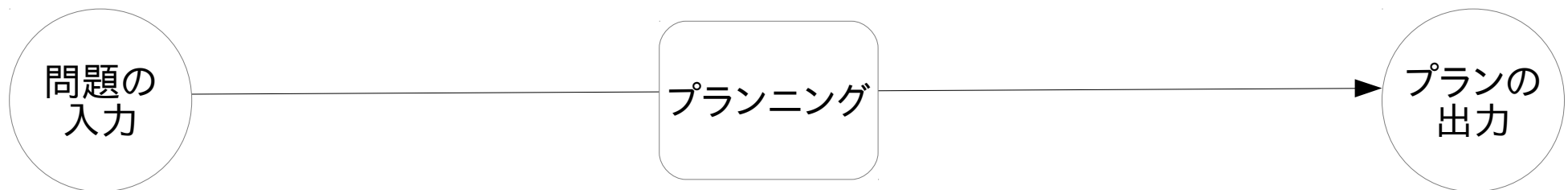
良いループ点=1周あたりのコストが小さいもの
どういうループ点はコストが小さいのか？

先行研究 (Asai 2013)より Automated Cyclic Planner

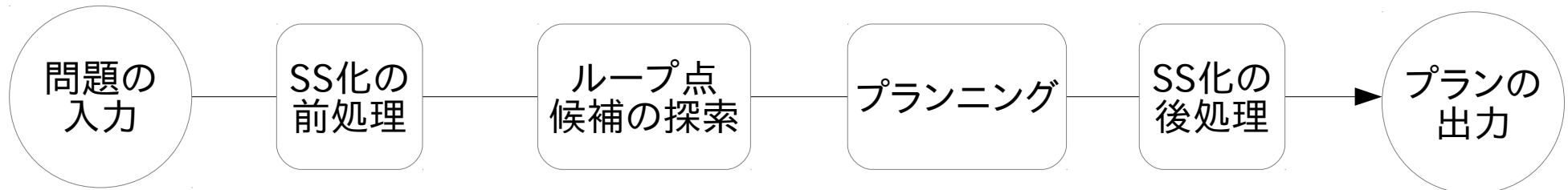
- Automated Cyclic Planner (ACP)

steady-stateモデルを利用したプランニングのプロセスを完全自動化

一般的なプランナ



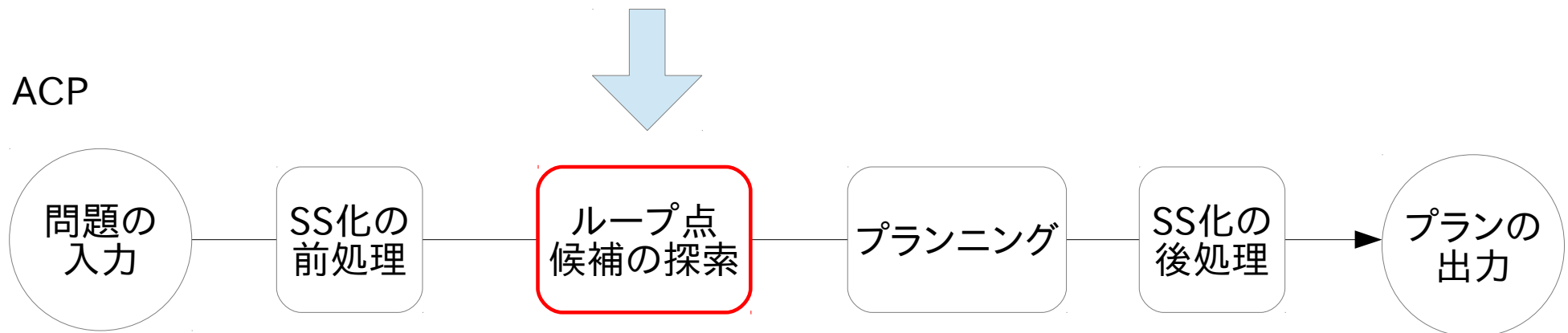
ACP



先行研究 (Asai 2013)より Automated Cyclic Planner

- Automated Cyclic Planner (ACP)
 - steady-stateモデルを利用したプランニングのプロセスを完全自動化

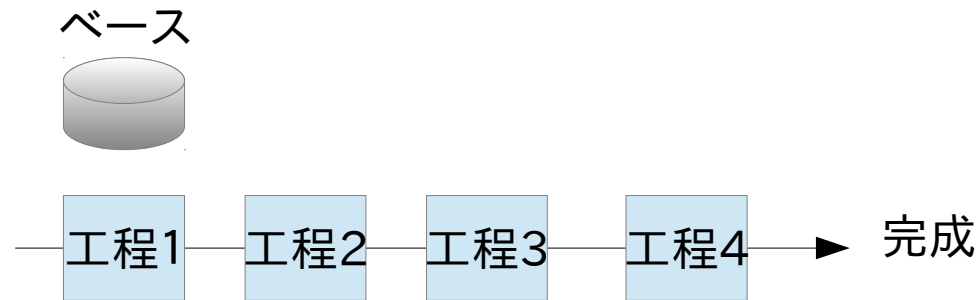
本研究ではこのプロセスでの問題を取り扱う



ACPの「ループ点」表現法

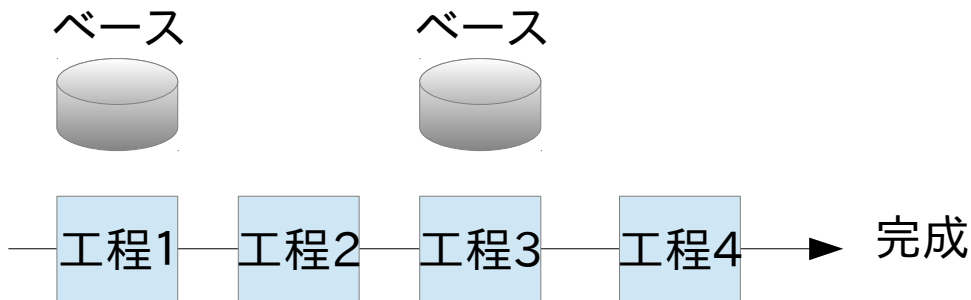
- ACPは一般的なプログラミングモデルの入力から「工程」を抽出する

ひとつのベースが完成するまでに通らなければいけない場所の順序

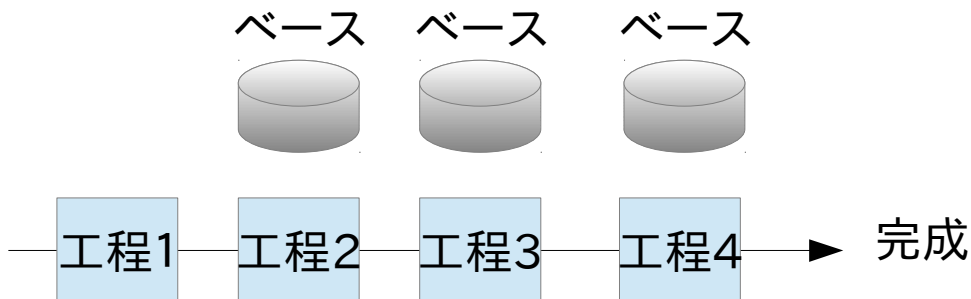


ACPの「ループ点」表現法

- ループ点は、抽出した各工程のどこにベースがある状態かによって表現する



工程1, 工程3にベースがある状態をループ
→ (1, 3)



工程2, 工程3, 工程4にベースがある状態をループ
→ (2, 3, 4)

ゆえに、ありうるループ点の候補数は
工程数が n のとき 2^n 通り

ACPのループ点探索法

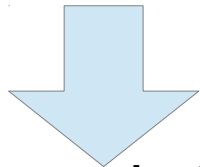
引き続き先行研究(Asai 2013)より

1. ありうるループ点の候補を全列挙
2. 実行不可能な候補を削除
 - デッドロックになる配置等
3. 残った候補から1ループあたりのコストが小さいループ点を探索
 - 探索法: ベースを1つ置く組合せ、2つ置く組合せ、3つ置く組合せ...を制限時間まで順に探索

ACPのループ点探索法

引き続き先行研究(Asai 2013)より

1. ありうるループ点の候補を全列挙
2. 実行不可能な候補を削除
 - デッドロックになる配置等
3. 残った候補から1ループあたりのコストが小さいループ点を探索
 - 探索法: ベースを1つ置く組合せ、2つ置く組合せ、3つ置く組合せ...を制限時間まで順に探索



- 本研究では3の探索過程に注目
 - もっと効率の良い探索法があるのではないか?

本研究の提案手法： ACPの探索アルゴリズムの改善

ACPのループ点探索に以下の機能を実装

1. 局所探索
2. 人の手による探索補助
(human-guided search)

目的: ACPのループ点探索における性能を向上させる

局所探索：山登り法

- 最急降下法
 - 近傍の中で最もコストの小さいノードへ移動
- ランダム降下法
 - 近傍をランダムな順に評価し、現在のノードより小さいコストを持つノードを発見し次第そこへ移動

いずれの方法でも局所最適解に到達した時点で探索を打ち切る

局所探索：近傍

探索はループ点単位

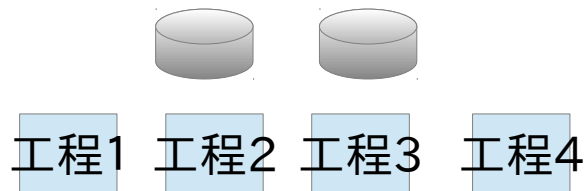
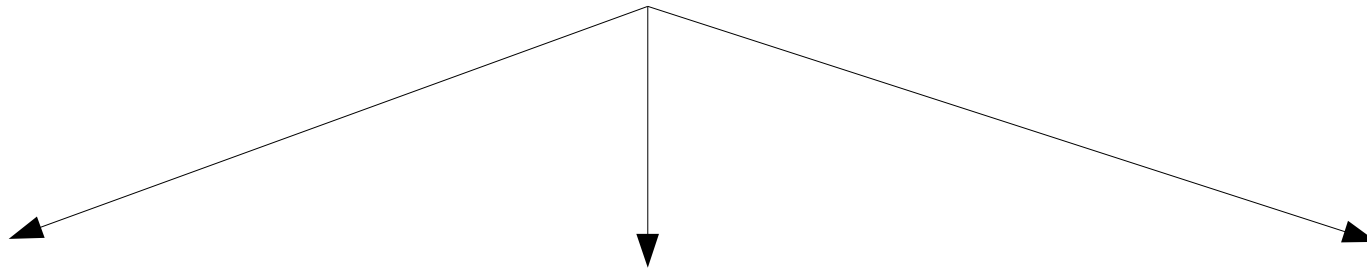
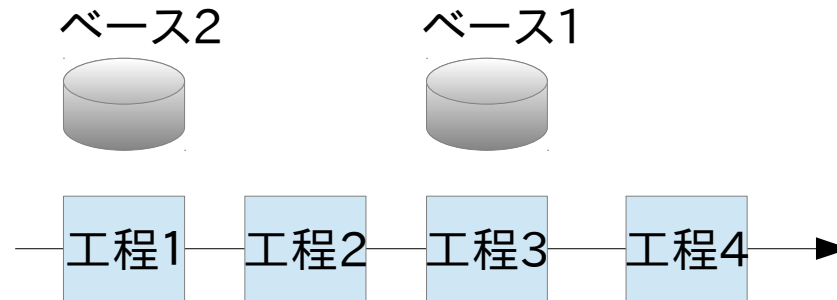
– 「あるループ点の近傍」をどう定義したらよいか？

局所探索における近傍の定義を3種用意して実験

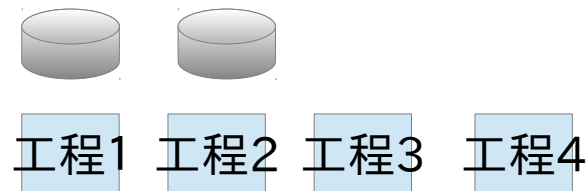
- 近傍1. ベースをひとつずらしたもの
- 近傍2. ベースをひとつ加えるか除いたもの
- 近傍 3. 近傍 1 \cup 近傍 2

近傍1. ベースをひとつずらす

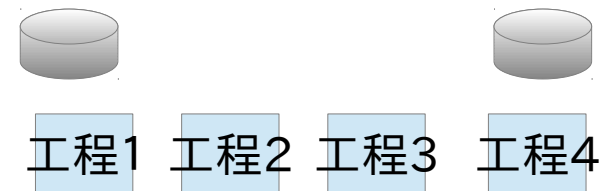
元のループ点



近傍



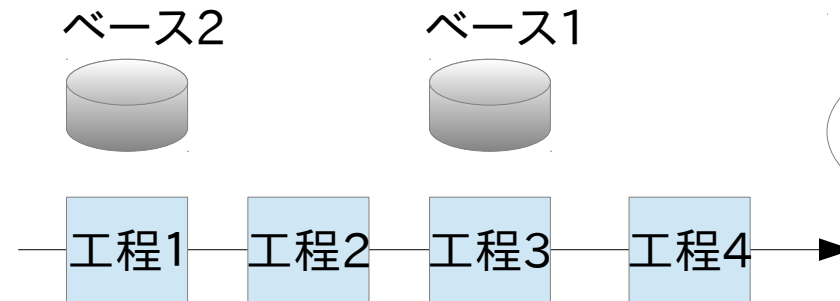
近傍



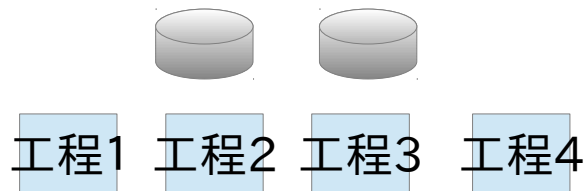
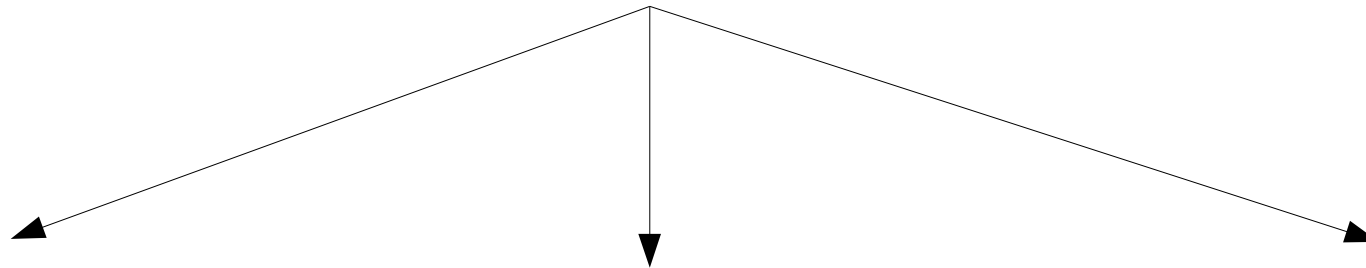
近傍

近傍1. ベースをひとつずらす

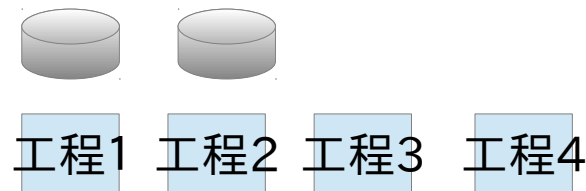
元のループ点



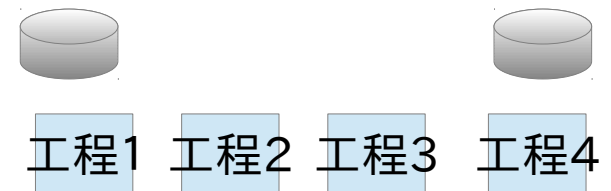
特徴: ベースの総数は
変わらない



近傍



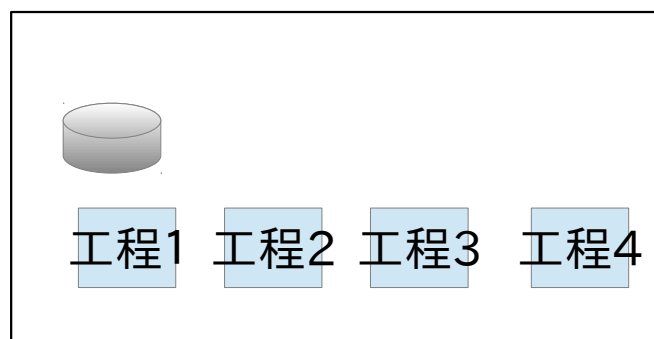
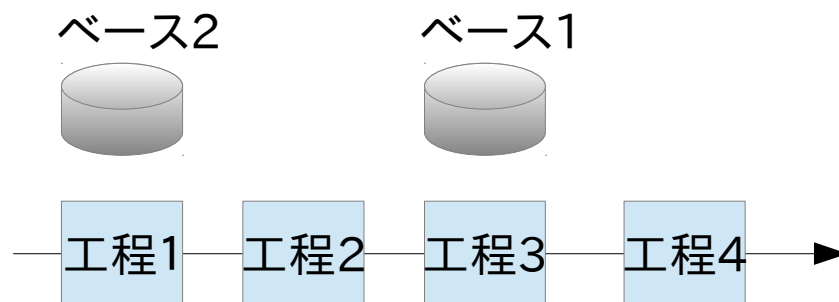
近傍



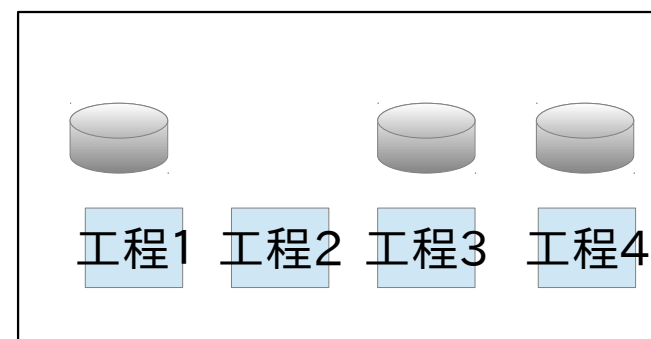
近傍

近傍2. ベースをひとつ加えるか除く

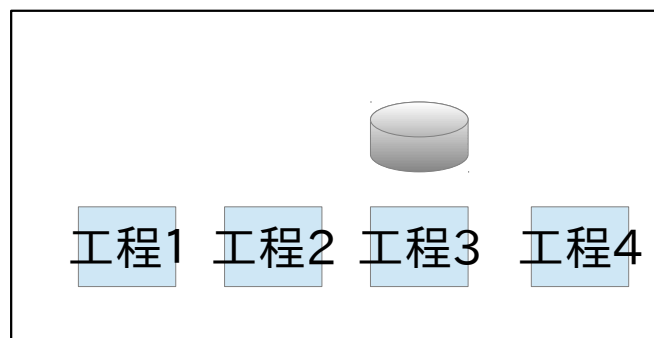
元のループ点



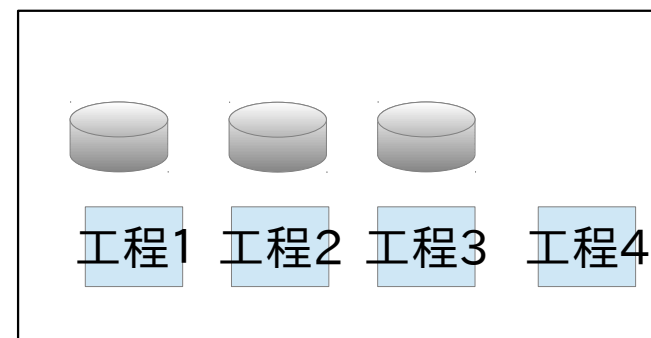
近傍



近傍



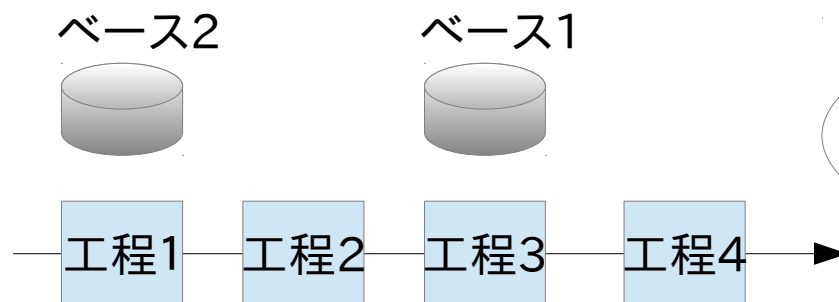
近傍



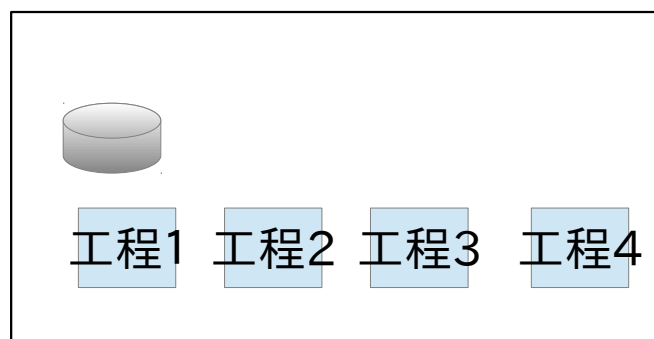
近傍

近傍2. ベースをひとつ加えるか除く

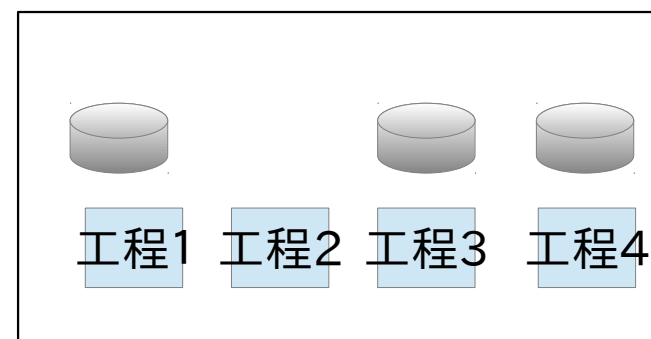
元のループ点



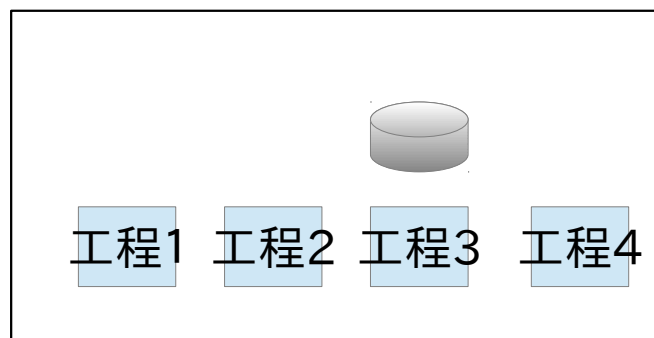
特徴: ベースの総数は必ず1増減する



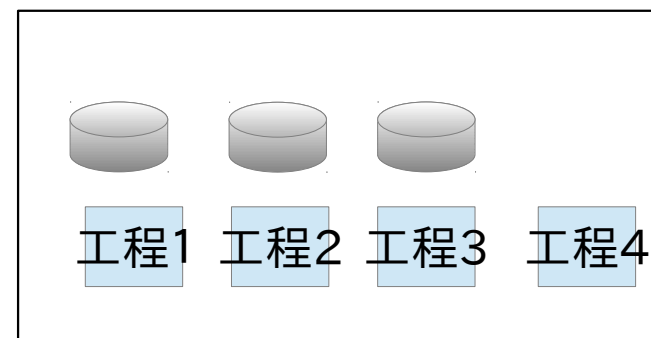
近傍



近傍



近傍



近傍

近傍3. 近傍1 \cup 近傍2

- 近傍1の定義で挙がるものと近傍2で挙がるもの
どちらも近傍とする

Human-guided search

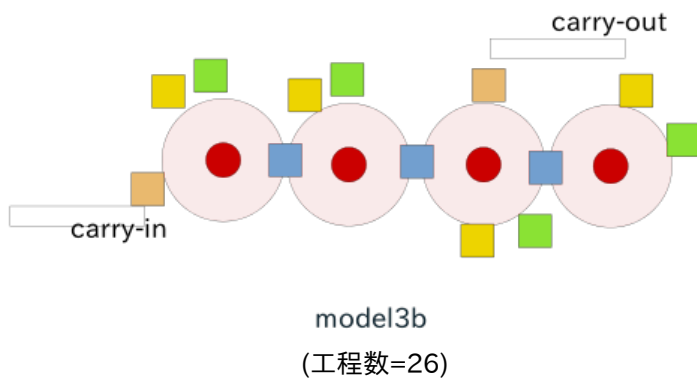
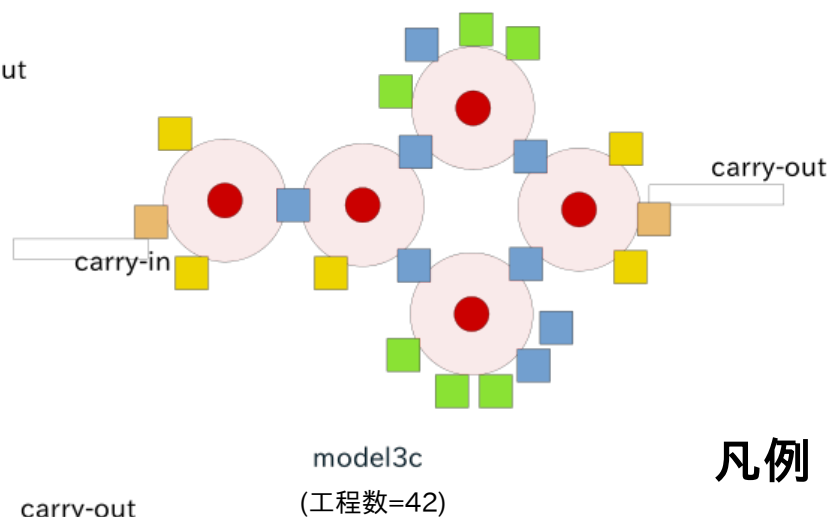
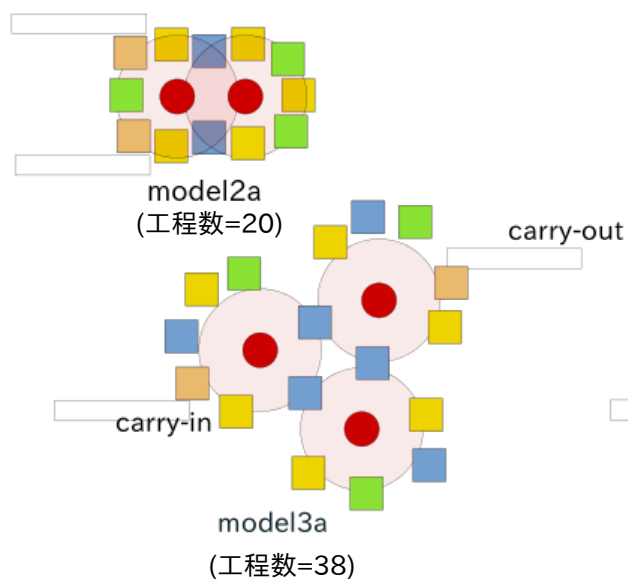
- 人間があらかじめモデルを見て効率の良さそうなループ点を決出し、ACPに伝える
- ACPはその点のコストを計算し、その点を出発点として局所探索を開始する

提案手法の処理の流れ

1. ループ点を指定してACPへ送信
2. ACPがそのループ点のコストを計算
3. 計算結果を受信
4. 結果をもとに局所探索
5. 再び新たなループ点を指定し1へ

実験：提案手法の効果検証

- 下図の4種+1種のセル生産モデルで実験

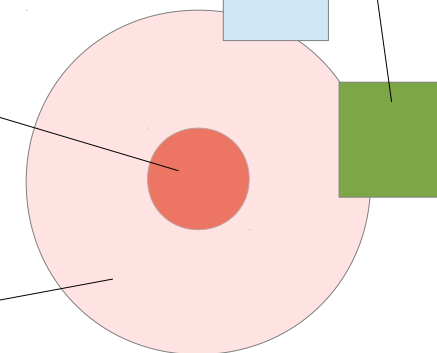


作業の過程で
ベースが置か
れる場所

凡例

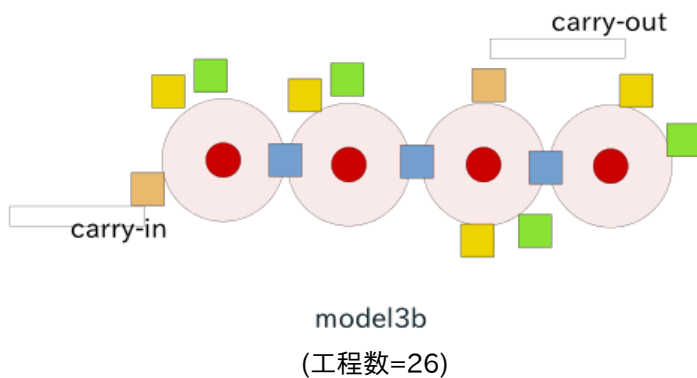
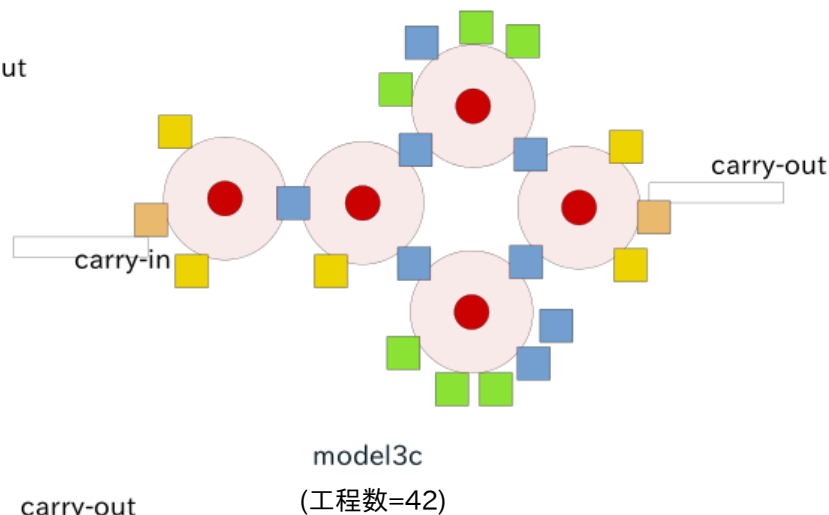
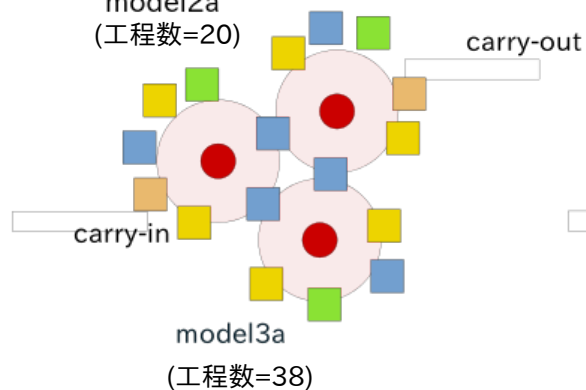
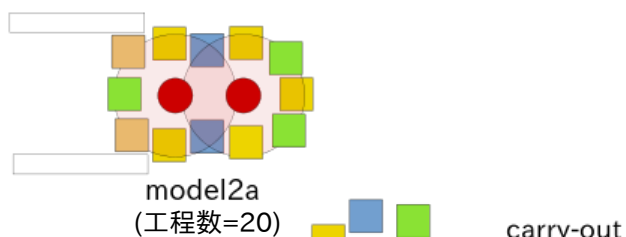
アーム

アームの届く範囲



実験：提案手法の効果検証

- 下図の4種+1種のセル生産モデルで実験



基本的に、アームと工程数が多いほど複雑な問題

実験設定：探索アルゴリズム

1. 最急降下法 + 人間による出発点の指定
 2. ランダム降下法 + 人間による出発点の指定
 3. 最急降下法 + ランダムな出発点の指定
 4. ランダム降下法 + ランダムな出発点の指定
 5. 完全ランダムサーチによる個別評価
- これらの探索法とACP(先行研究の実験結果)との間で、発見した最適コストの値を比較
 - いずれも実験時間は30分 ACP60分
 - ただし3, 4, 5は乱数への依存が大きいため5回ずつ探索を行いその平均値を求めた
 - 局所探索の近傍には3種類の定義を用意してそれぞれ実験

実験結果 (近傍1)

各アルゴリズムの発見した最適コスト

	最急降下 +人間	ランダム降下 +人間	最急降下 +ランダム	ランダム降下 +ランダム	完全ランダム	ACP単独 (既存手法)
model2a (工程数:20)	49.0	49.0	73.0	81.0	61.0	77.0
model2b (工程数:10)	61.0	61.0	61.0	61.0	61.0	61.0
model3a (工程数:38)	154.0	154.0	133.8	151.8	144.8	141.0
model3b (工程数:26)	56.0	56.0	64.0	70.4	55.4	63.0
model3c (工程数:42)	108.0	108.0	154.0	156.4	183.4	145.0

赤太字: 各モデル中で最良の結果
赤字: 既存手法より良い結果のもの

- すべてのモデルで、提案手法のいずれかが最も良いコストを出した

実験結果 (近傍2)

各アルゴリズムの発見した最適コスト

	最急降下 +人間	ランダム降下 +人間	最急降下 +ランダム	ランダム降下 +ランダム	完全ランダム	ACP単独 (既存手法)
model2a (工程数:20)	54.0	49.0	69.0	59.0	61.0	77.0
model2b (工程数:10)	61.0	61.0	61.0	61.0	61.0	61.0
model3a (工程数:38)	123.0	131.0	139.8	147.8	144.8	141.0
model3b (工程数:26)	56.0	56.0	64.6	75.6	55.4	63.0
model3c (工程数:42)	99.0	99.0	119.2	197.6	183.4	145.0

赤太字: 各モデル中で最良の結果
赤字: 既存手法より良い結果のもの

- すべてのモデルで、提案手法のいずれかが最も良いコストを出した

実験結果 (近傍3)

各アルゴリズムの発見した最適コスト

	最急降下 +人間	ランダム降下 +人間	最急降下 +ランダム	ランダム降下 +ランダム	完全ランダム	ACP単独 (既存手法)
model2a (工程数:20)	49.0	49.0	80.4	71.8	61.0	77.0
model2b (工程数:10)	61.0	61.0	61.0	61.0	61.0	61.0
model3a (工程数:38)	129.0	131.0	128.6	132.4	144.8	141.0
model3b (工程数:26)	56.0	56.0	66.0	69.6	55.4	63.0
model3c (工程数:42)	107.0	100.0	155.8	154.2	183.4	145.0

赤太字: 各モデル中で最良の結果
赤字: 既存手法より良い結果のもの

- すべてのモデルで、提案手法のいずれかが最も良いコストを出した

考察

- 特にサイズの大きい問題において局所探索や人間による介入は、既存手法の単純な列挙戦略に大した大きな性能向上を示した
- 中程度のサイズの問題では完全ランダムサーチが良い結果を示す傾向が見られた
- 問題サイズが小さすぎる場合、どのような探索法でもすぐに最適解が発見されるので効果は薄い

結論

- 本研究ではループの概念を用いた周期的プランニングに対して、自動的に効率の良いループの設定を求めるアルゴリズムを提案した
- 単純すぎない多くの問題において、提案手法は従来の探索手法よりも良い結果を示した
- 「セル生産システムのためのプランニングの性能向上をめざす」という本研究の目的は達成された
- ACPはセル生産方式だけでなく、流れ作業による生産システム全般に対して汎用的に適用可能
 - 従って本研究の手法も汎用的な有用性を持つことが期待される