ＧＡを用いた
都市間鉄道網計画策定支援システムの
計算効率化に関する研究

波床　正敏

（大阪産業大学 ）

★ はじめに

★ 研究の背景

- GAを使った 都市間鉄道網分析に 時間かかりすぎる
  - 例えば、九州を対象として、100～200時間 [数日]
    (2007年時点)
- 全国対象では、2,000時間 [3ヶ月]でも終わらない
  - 条件を変えて様々な計算したい
  - 適合度の計算方法を 高度化したい
  - 大型計算機を借りると 数百万円～数千万円 ?!

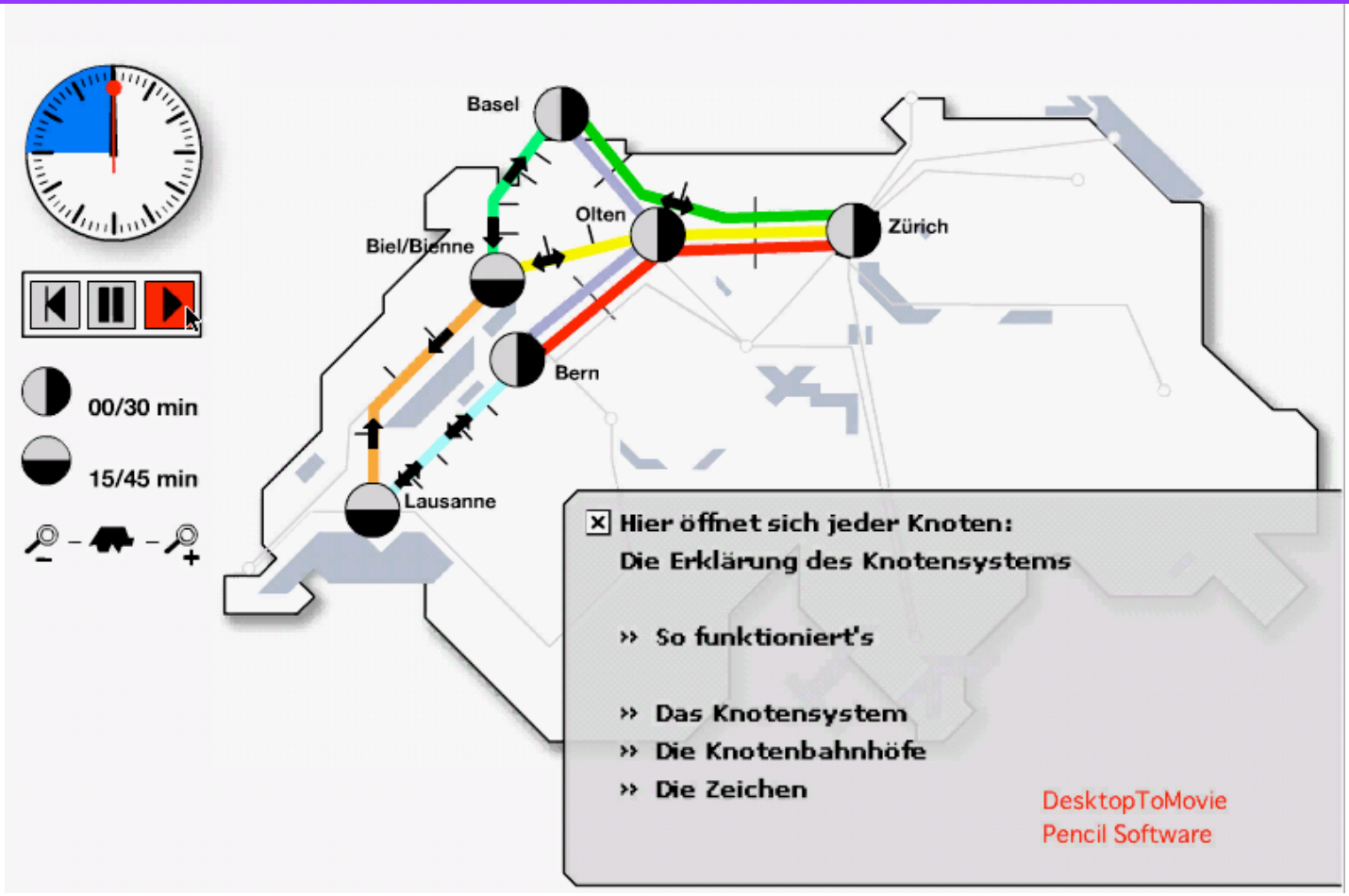
★ 研究の目的

- 計算時間を大幅に短くしたい
  - できれば、[当初比]10倍速くらいにならないか？
- GAにおける汎用的な知見 と 非汎用的な知見
  - GAそのものが研究目的ではない場合、どうすれば効率よく計算できるか

★ 計算システムの概要

～何を計算しようとしているのか～

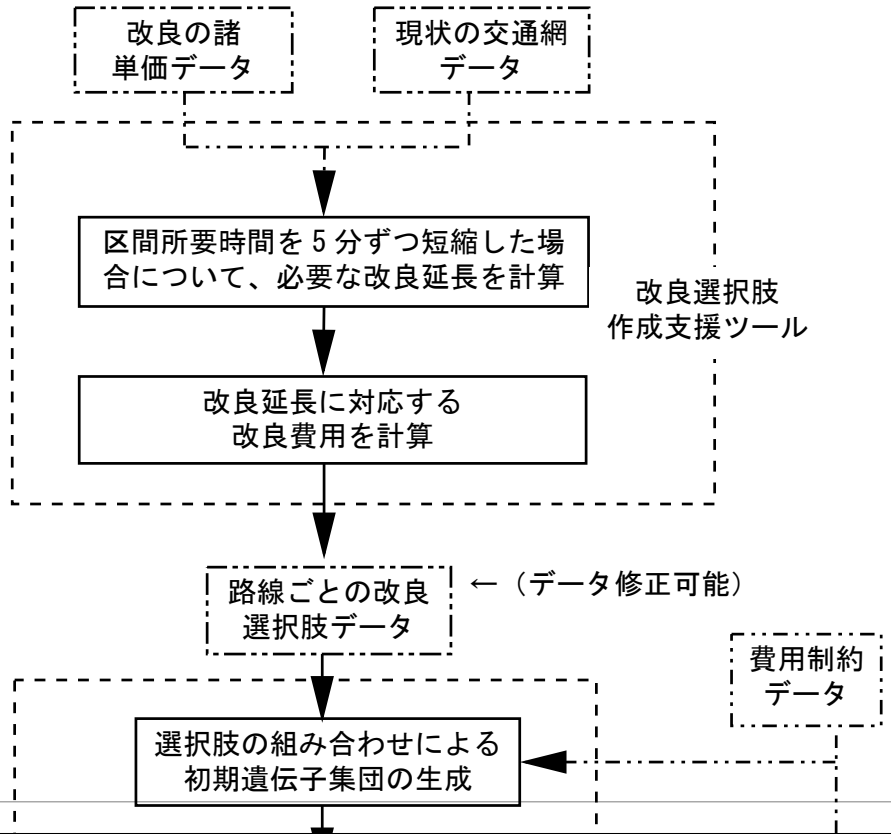
★ スイスの鉄道政策 Rail 2000



★ 日本に導入するには

- どの路線の 所要時間を
- どれだけ 短縮するか
  - つまり、どの路線に 何億円投入するか？
  - 費用の制約は？
- 列車の出発タイミングは？

システムの概要 (効率改善前)



キロあたり路線改良単価設定

表１　在来線の改良単価設定

改良前の条件				改良後の条件				キロ単価億円	参考事例
単複	電化	振子	表定速度km/h	単複	電化	振子	表定速度km/h		
単	電	振	51.5	単	電	振	69.7	0.64	紀勢線白浜以南高速化試算
〃	〃	〃	51.5	〃	〃	非	96.8	13.45	同、ミニ新幹線化(路線付替)
複	〃	〃	85.5	複	〃	振	92.0	0.13	阪和線高速化試算
〃	〃	〃	86.7	〃	〃	〃	98.9	0.81	紀勢線白浜以北高速化試算
〃	〃	〃	86.7	〃	〃	非	115.1	12.36	同、ミニ新幹線化(路線付替)
〃	〃	〃	94.6	〃	〃	振	106.2	5.60	高尾－甲府 130km/h 化試算
〃	〃	〃	94.6	〃	〃	〃	123.1	20.13	同、160km/h 化試算
単	非	非	46.3	単	非	非	56.8	0.21	津山線高速化事業

キロあたり建設単価設定

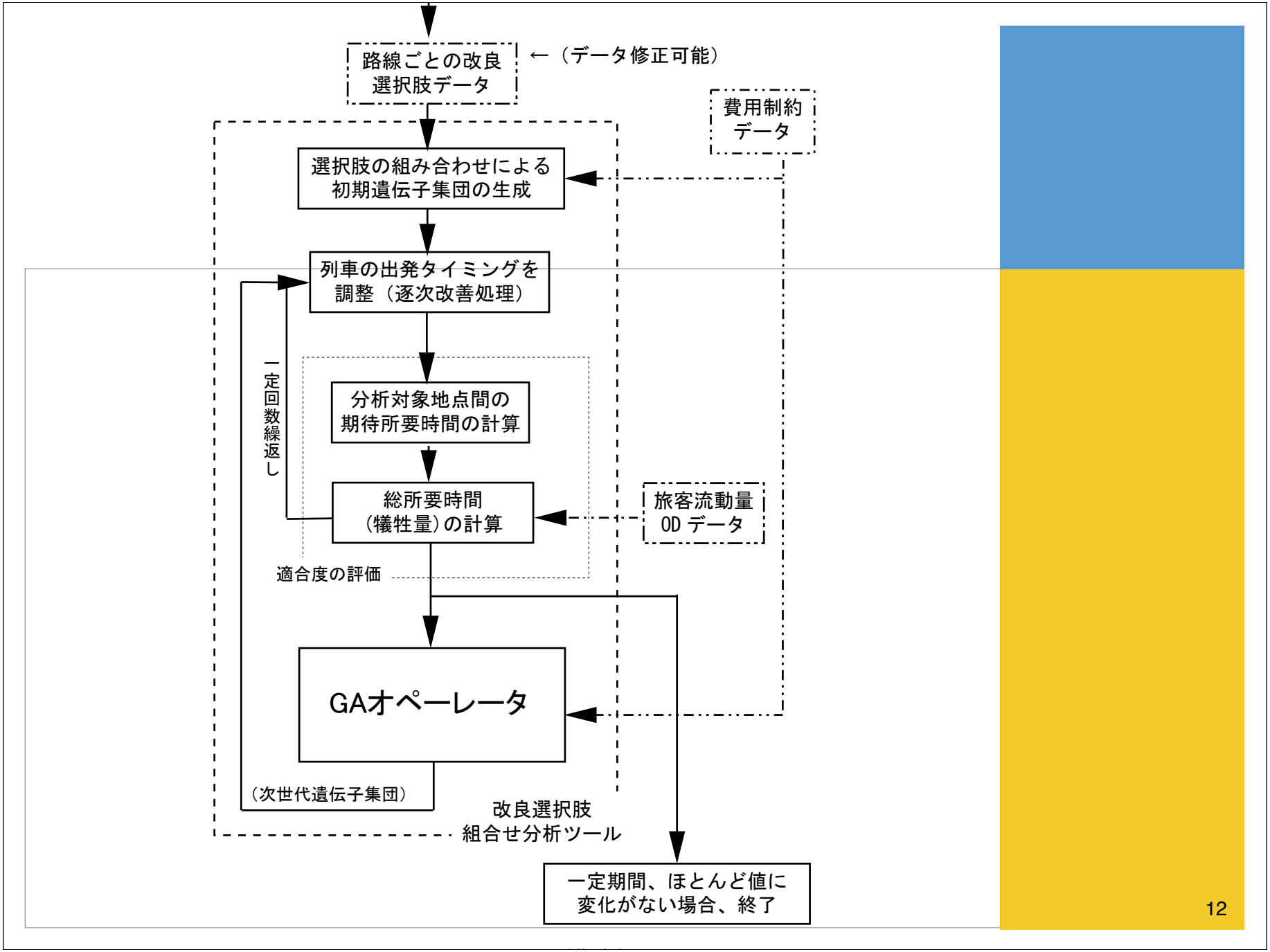
表２　新線建設・新幹線速度向上費用の単価設定

	億円 /Km	表定速度(Km/h)	備考
新線 130km/h	29.30	91.9	複線電化[延長 10.0km 以上]
新線 160km/h	35.95	113.1	複線電化[延長 12.3km 以上]
新線 260km/h	58.18	213.3	フル規格新幹線[20.0Km 以上]
新線 500km/h	188.88	453.9	リニア新幹線
新幹線高速化	0.78	+10.1	最高速度向上幅 10km/h あたり

(例) 久大本線(久留米-日田)の改良選択肢

表－１　久大本線(久留米－日田)の改良選択肢(例)

番	分	億円	改良後の状態	改良長(km)	参考事例
1	48	0	単線,非電化,非振子	0	基本(現状)
2	43	4.3	単線,非電化,振子	44.3	中村線高速化事業
3	38	16.1	単線,非電化,振子	39.0	智頭急行高速化試算
4	33	81.9	単線,電化,振子	43.5	宮福線高速化試算
5	28	1504.5	複線,電化,非振子	41.8	160Km/h 運転新線
6	14	2769.4	複線,電化,非振子	47.6	260Km/h 運転新線
7	12	2917.4	複線,電化,非振子	47.6	300Km/h 運転新線
8	10	3102.1	複線,電化,非振子	47.6	350Km/h 運転新線


路線ごとの改良
選択肢データ
←（データ修正可能）
費用制約
データ
選択肢の組み合わせによる
初期遺伝子集団の生成
列車の出発タイミングを
調整（逐次改善処理）
一定回数繰返し
分析対象地点間の
期待所要時間の計算
総所要時間
（犠牲量）の計算
旅客流動量
OD データ
適合度の評価
GAオペーレータ
（次世代遺伝子集団）
改良選択肢
組合せ分析ツール
一定期間、ほとんど値に
変化がない場合、終了

	選択	番号	所要時間(分)	順方向発時刻(毎時,分)	逆方向発時刻(毎時,分)	費用(億円)
:	:	:	:	:	:	:
日豊線 大分 ↓↑ 佐伯		1	54	−	−	0.0
		2	53	−	−	1.9
		3	48	−	−	11.6
	○	4	43	25	0	72.3
		5	38	−	−	329.4
		6	19	−	−	3775.9
		7	16	−	−	3977.7
		8	13	−	−	4229.5
日豊線 佐伯 ↓↑ 延岡		1	58	−	−	0.0
		2	53	−	−	4.4
		3	48	−	−	14.7
		4	43	−	−	63.8
	○	5	38	15	10	120.8
		6	33	−	−	1942.6
		7	17	−	−	3397.7
		8	13	−	−	3669.9
		9	12	−	−	3805.9
日豊線 延岡 ↓↑ 宮崎		1	68	−	−	0.0
		2	63	−	−	10.1
		3	58	−	−	20.3
	○	4	53	0	0	145.3
		5	48	−	−	445.2
		6	24	−	−	4869.7
		7	20	−	−	5130.0
		8	17	−	−	5454.7
:	:	:	:	:	:	:

評価指標（適合度）：Σ期待所要時間×幹線旅客純流動量

期待所要時間：乗り継ぎの利便性や速度などを総合的に反映

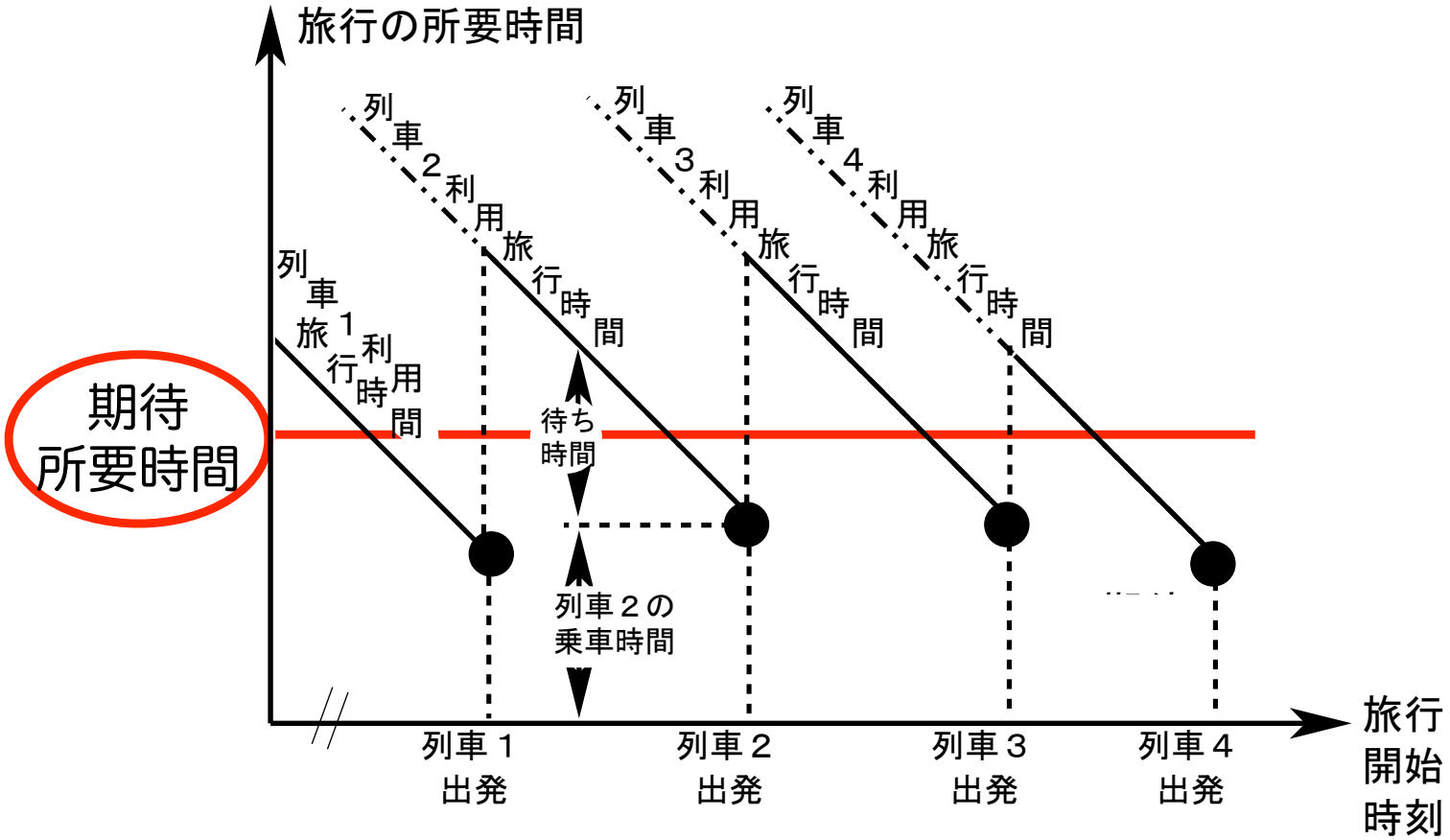
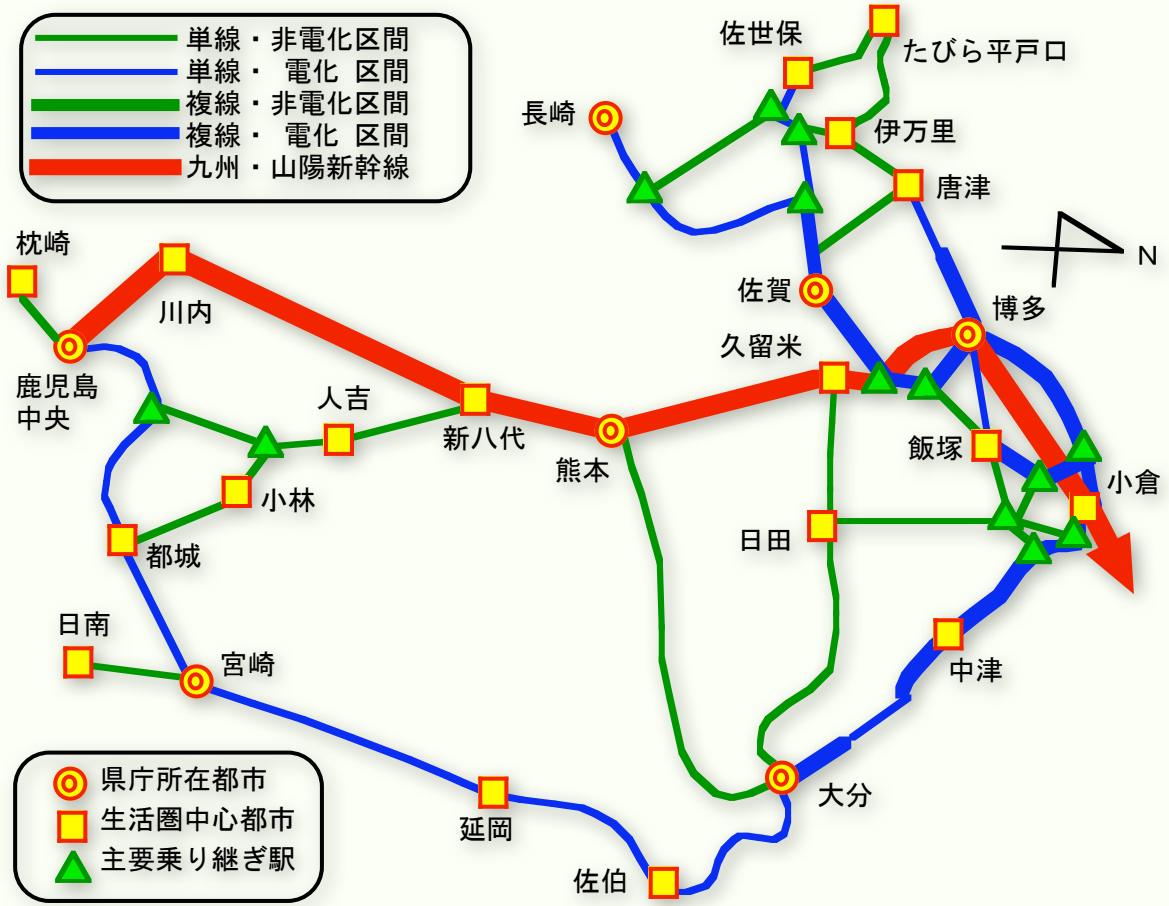


図２　期待所要時間と実最短所要時間の考え方

分析対象都市(駅名)
鉄道ネットワーク



乗り継ぎ

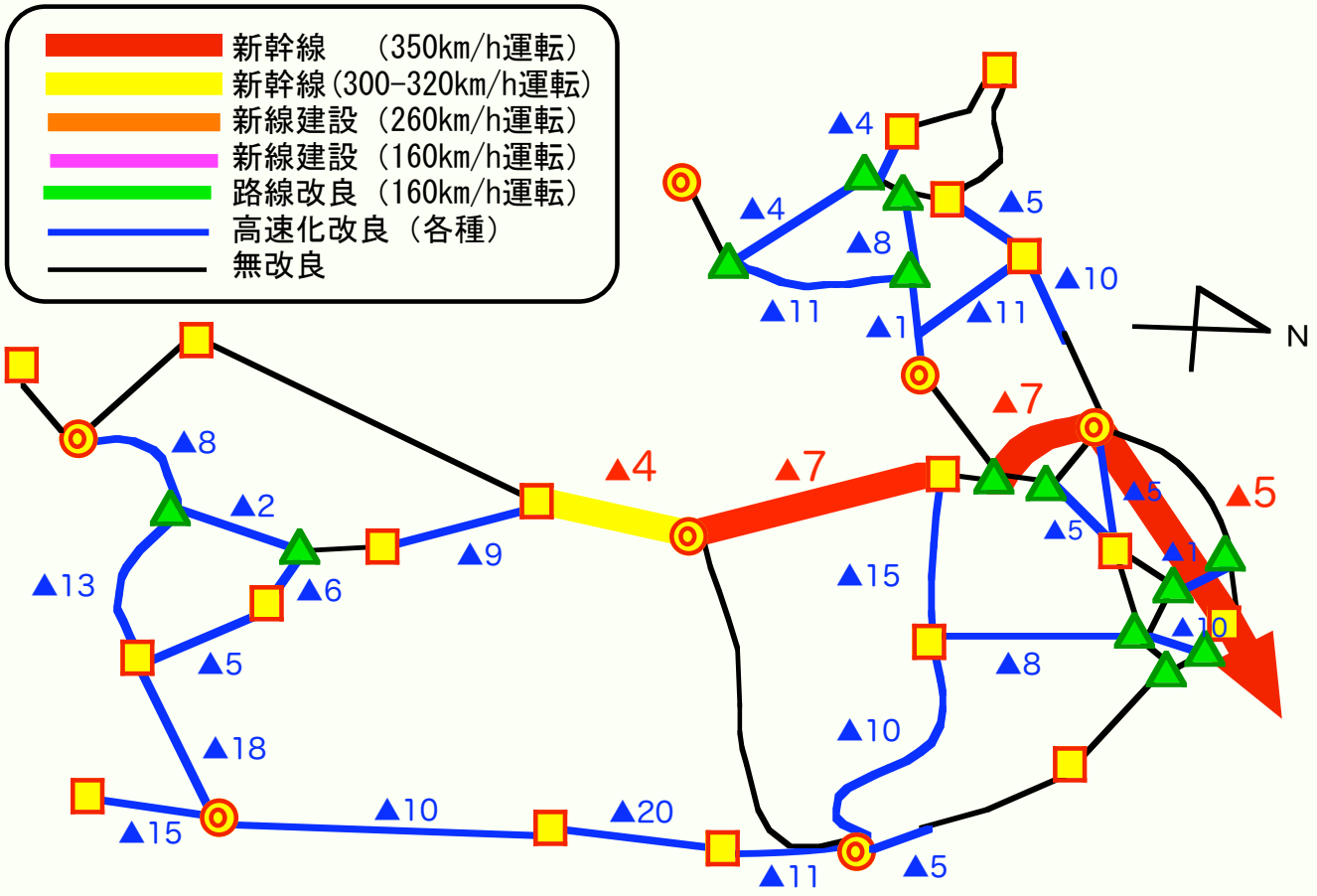
新幹線相互：５分

在来線相互：５分

新幹線と
在来線：７分

計算例：
費用制約：2,500億円
九州幹線鉄道ネットワーク

GAによる
計算結果



★ システムの改良

システムの改良ポイント等

① 評価値計算の並列処理

② 集団サイズ (評価対象の個体数) の最適化

③ パラメタ(交叉率,突然変異率)の検討

④ 打切り条件の検討 (どこまで計算するか)

⑤ 試行回数の検討 (何回繰り返せば、いい結果が得られるか)

⑥ 「早熟な収束」対策 (積極的な活用)

(⑦ コーディング上の工夫)

システムの改良ポイント等

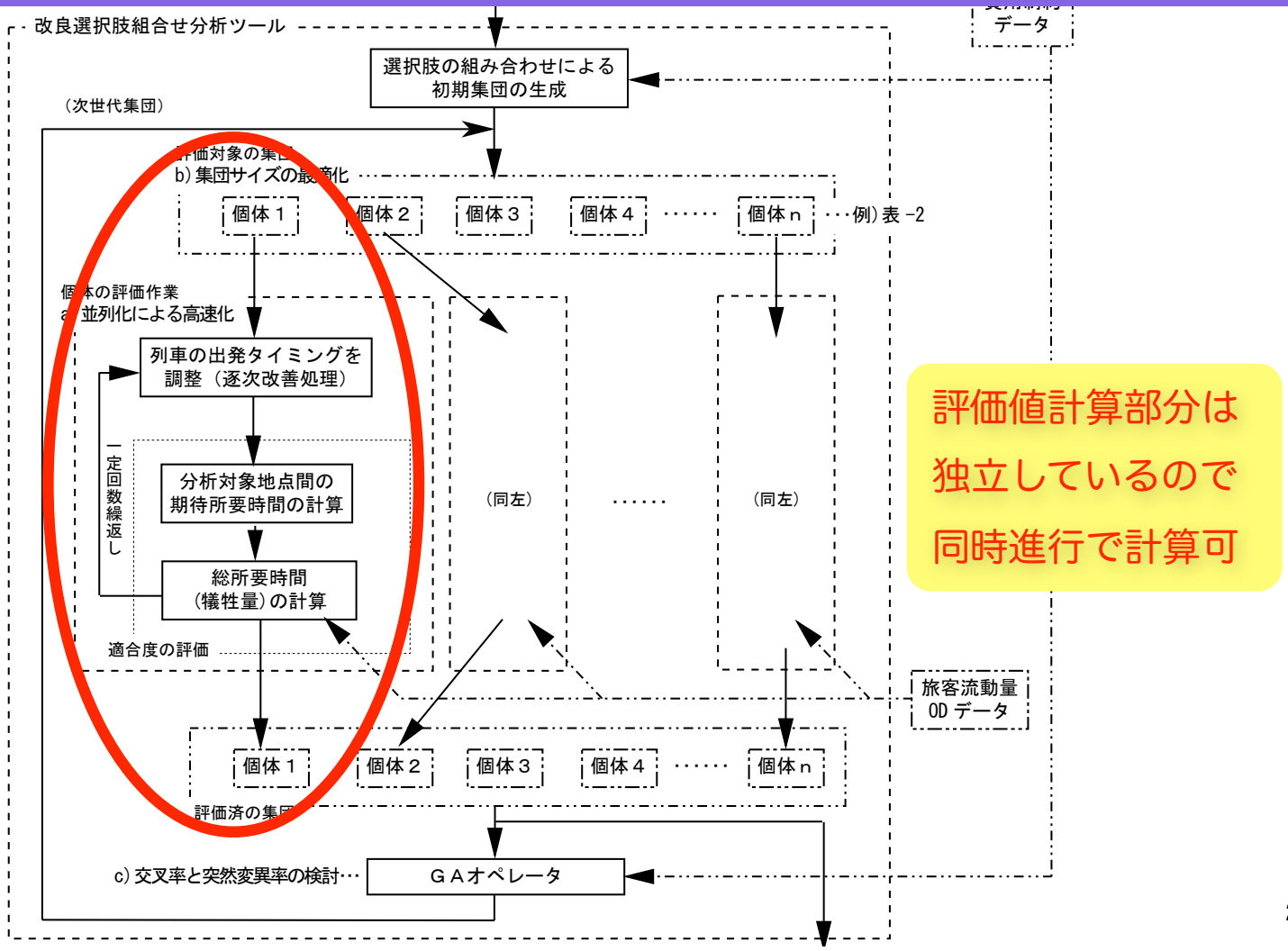
基本設定

交叉率　　：70%
突然変異率　：5%
エリート戦略併用 (集団サイズの10%)
集団サイズ　：ビット数で最適化
並列化　　：4 並列
終了条件　　：100世代あたりの適合度の変化率 < 0.1%
繰り返し　　：10回繰り返して平均値や最良値を求める

表－３ 使用した計算機の基本スペック

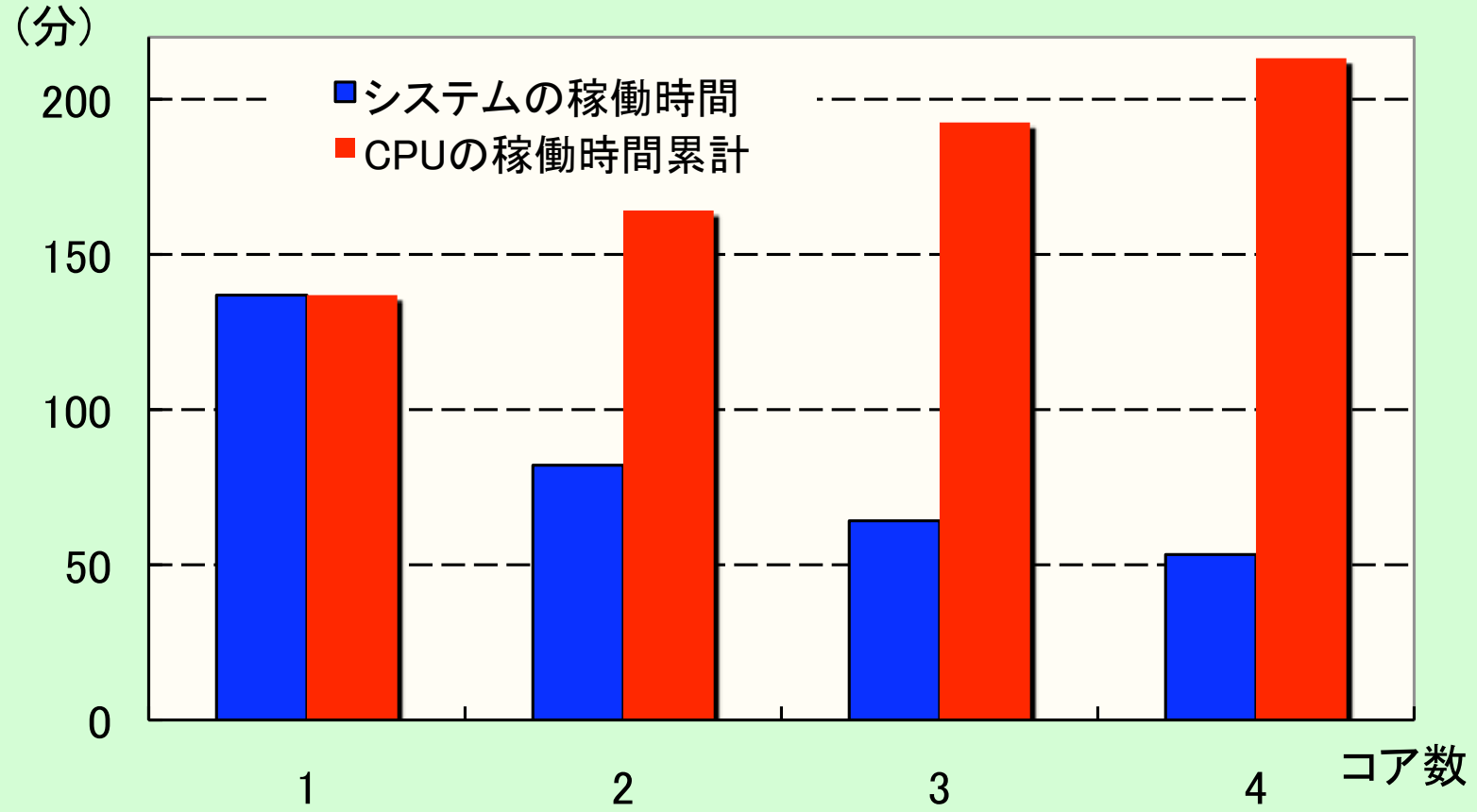
CPU	Intel Core2 Quad Q9650 又は Q9550 (Clock = 424MHz×8.5 = 3.6GHz)
Chipset	P45, P35, G33
Memory	DDR2 SDRAM 1GB×2 (Dual Channel) (Clock = 424MHz×2.0 = 848MHz)
OS	Windows Vista (64bit 版)
Compiler	Intel Visual Fortran 9.1 (64bit 版) Core2 に最適化

① 評価値計算の並列処理



評価値計算部分は
独立しているので
同時進行で計算可

① 評価値計算の並列処理



図−６　使用コア数による処理時間の変化

② 集団サイズ (評価対象の個体数) の最適化

- 大きな集団サイズ
  - 良好な結果を得られる可能性がある
    - 多峰性関数＆遺伝子長大
  - 計算時間がかかる
- 小さな集団サイズ
  - 早熟な収束の可能性高まる
  - 計算時間は短い

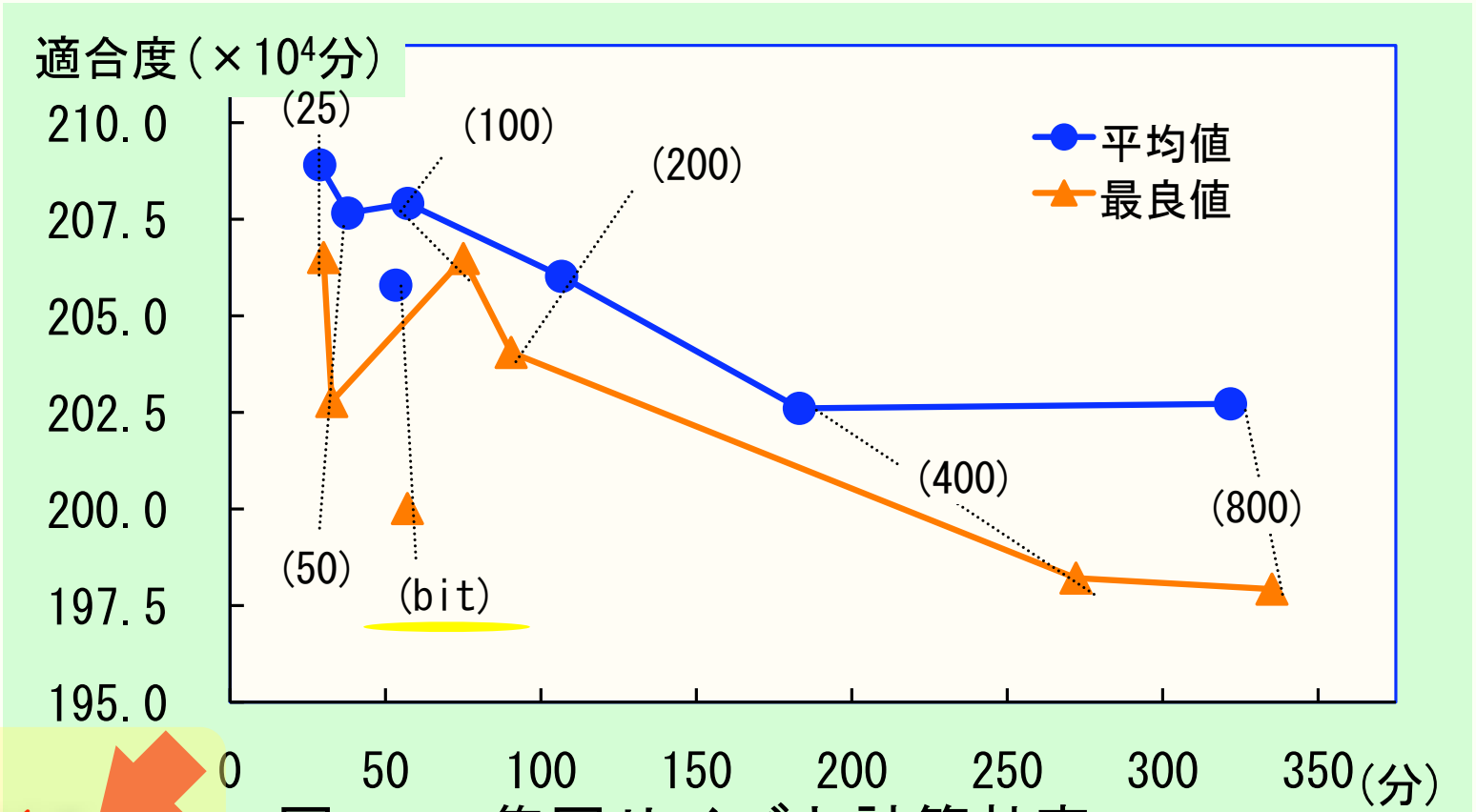
	選択	番号	所要時間(分)	順方向発時刻(毎時,分)	逆方向発時刻(毎時,分)	費用(億円)
:	:	:	:	:	:	:
日豊線 大分 ↓↑ 佐伯		1	54	−	−	0.0
		2	53	−	−	1.9
		3	48	−	−	11.6
	○	4	43	25	0	72.3
		5	38	−	−	329.4
		6	19	−	−	3775.9
		7	16	−	−	3977.7
		8	13	−	−	4229.5
日豊線 佐伯 ↓↑ 延岡		1	58	−	−	0.0
		2	53	−	−	4.4
		3	48	−	−	14.7
		4	43	−	−	63.8
	○	5	38	15	10	120.8
		6	33	−	−	1942.6
		7	17	−	−	3397.7
		8	13	−	−	3669.9
		9	12	−	−	3805.9
日豊線 延岡 ↓↑ 宮崎		1	68	−	−	0.0
		2	63	−	−	10.1
		3	58	−	−	20.3
	○	4	53	0	0	145.3
		5	48	−	−	445.2
		6	24	−	−	4869.7
		7	20	−	−	5130.0
		8	17	−	−	5454.7
:	:	:	:	:	:	:

例えば、
こんな高価な選択肢は
どの個体にも出てこない

② 集団サイズ (評価対象の個体数) の最適化

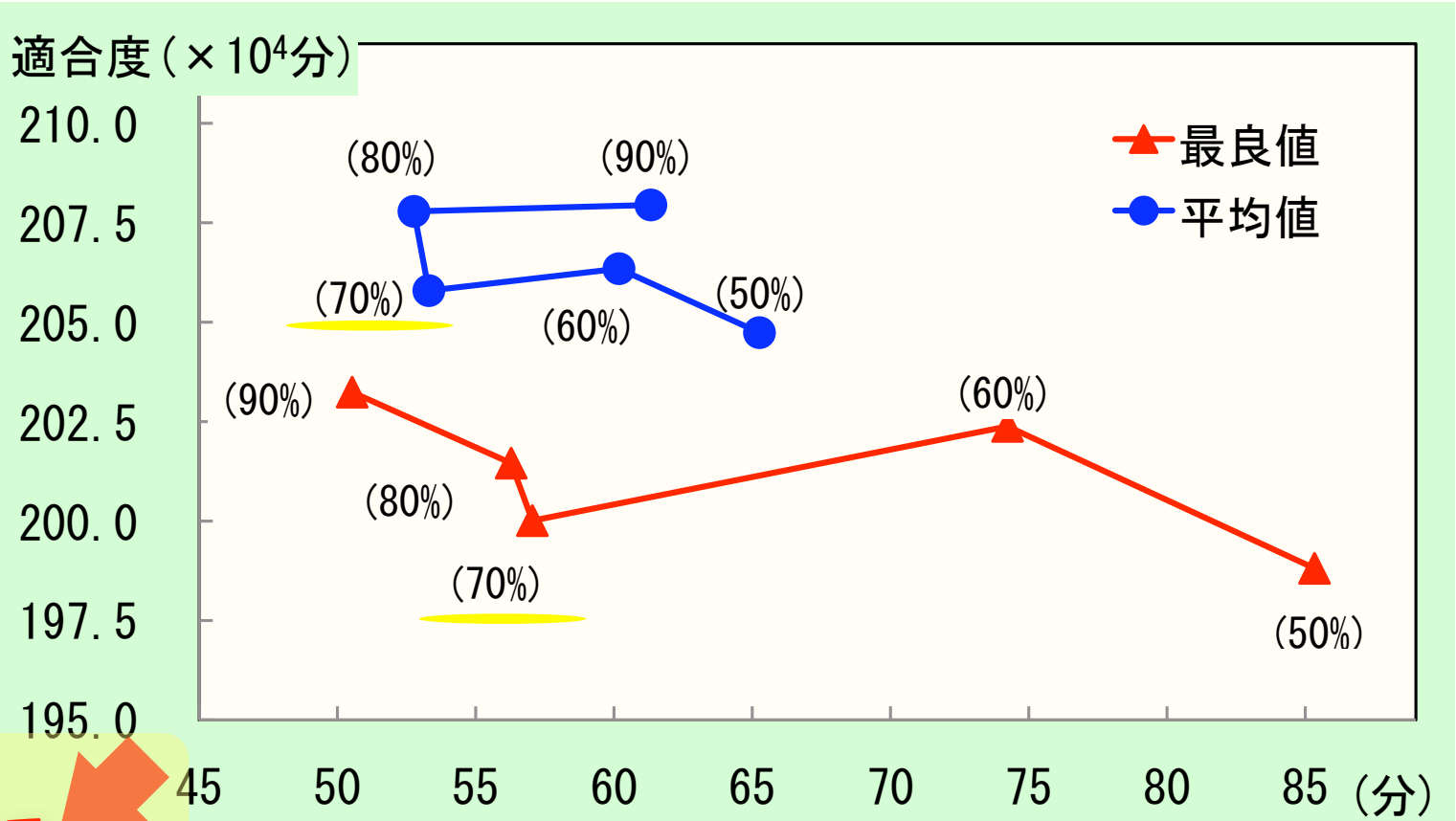
- 集団サイズ最適化
  - 実際に使っている選択肢数→ビット換算
  - 集団サイズ←総ビット数
  - 動的に集団サイズを変更

② 集団サイズ (評価対象の個体数) の最適化



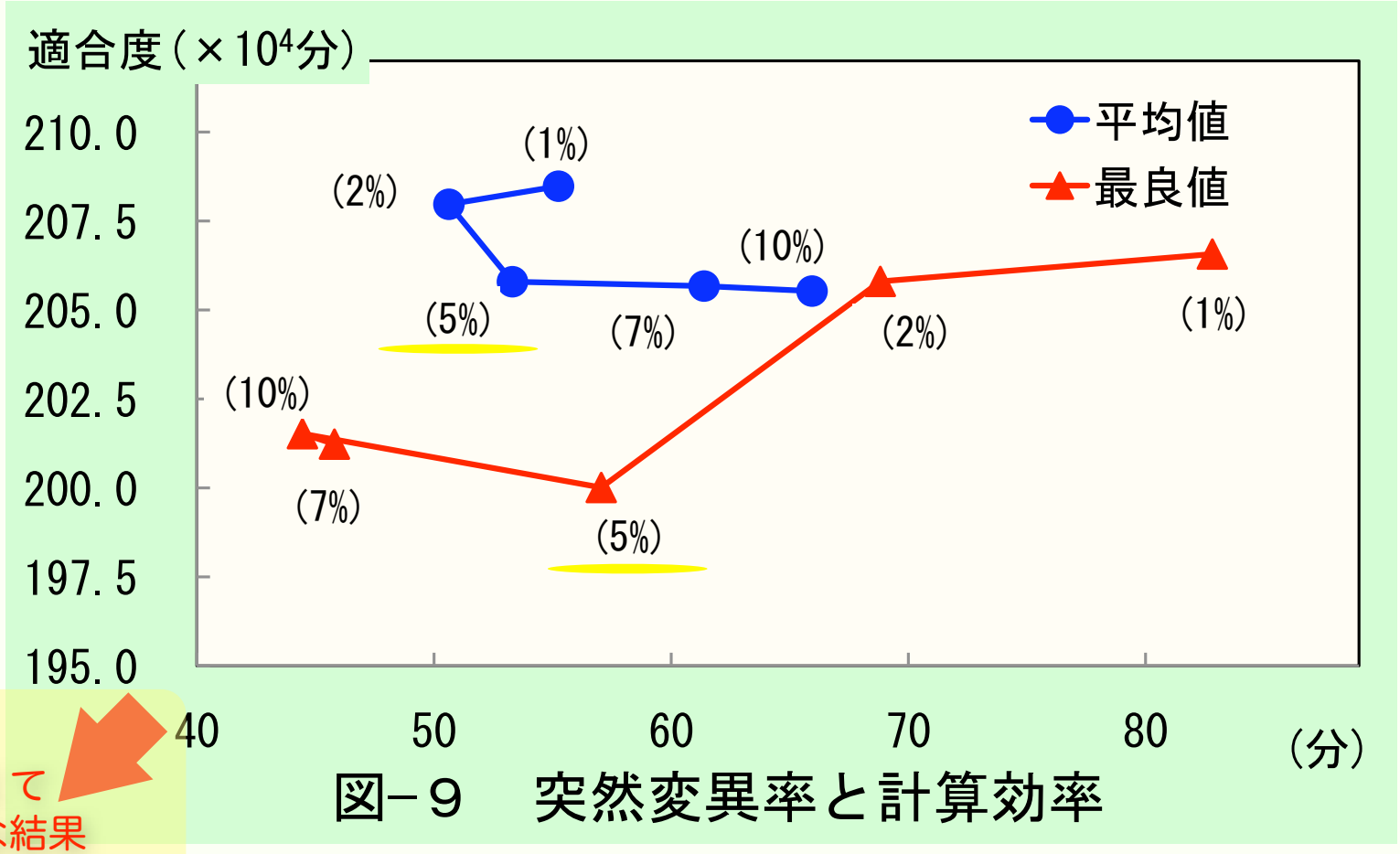
図−7　集団サイズと計算効率

③ パラメタ(交叉率,突然変異率)の検討



図−８　交叉率と計算効率

③ パラメタ(交叉率,突然変異率)の検討



図−９　突然変異率と計算効率

④ 打切り条件の検討 (どこまで計算するか)

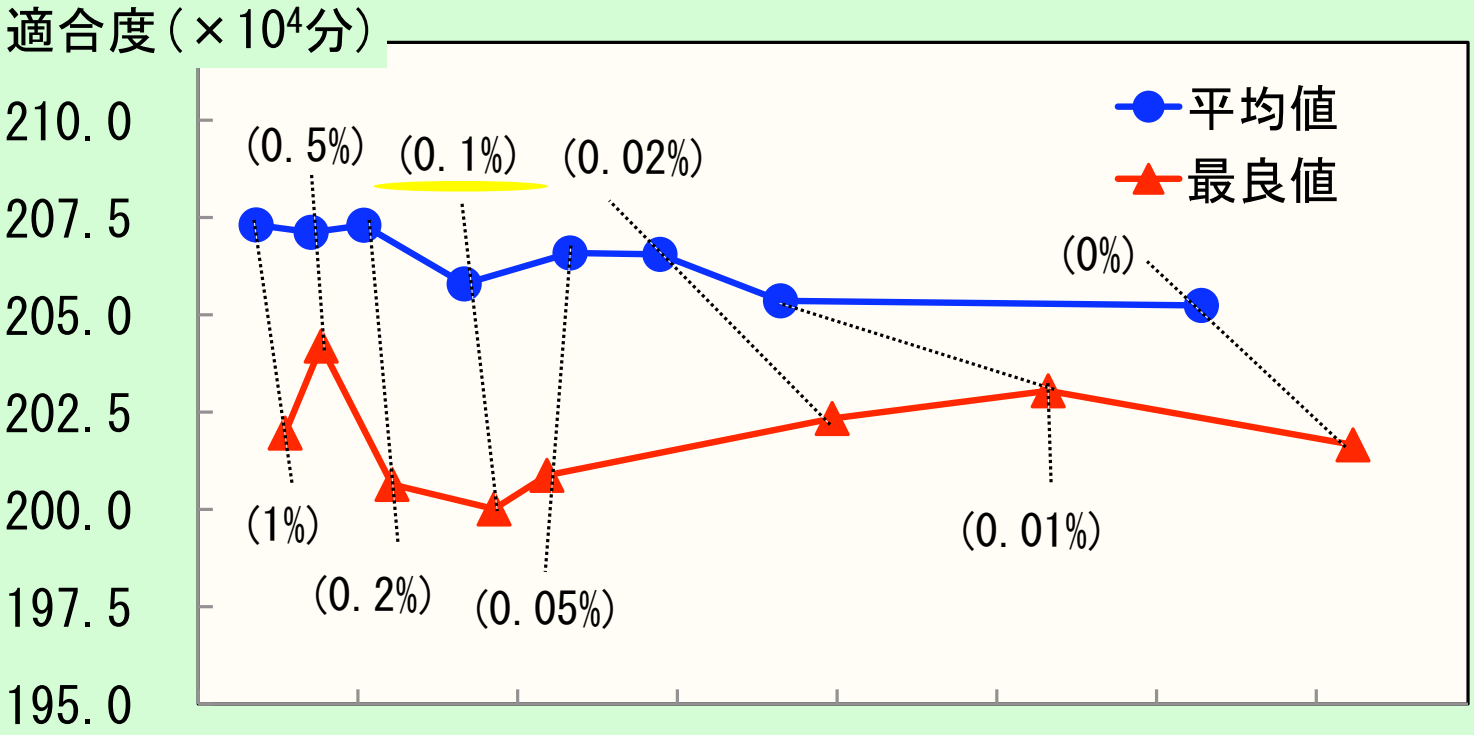
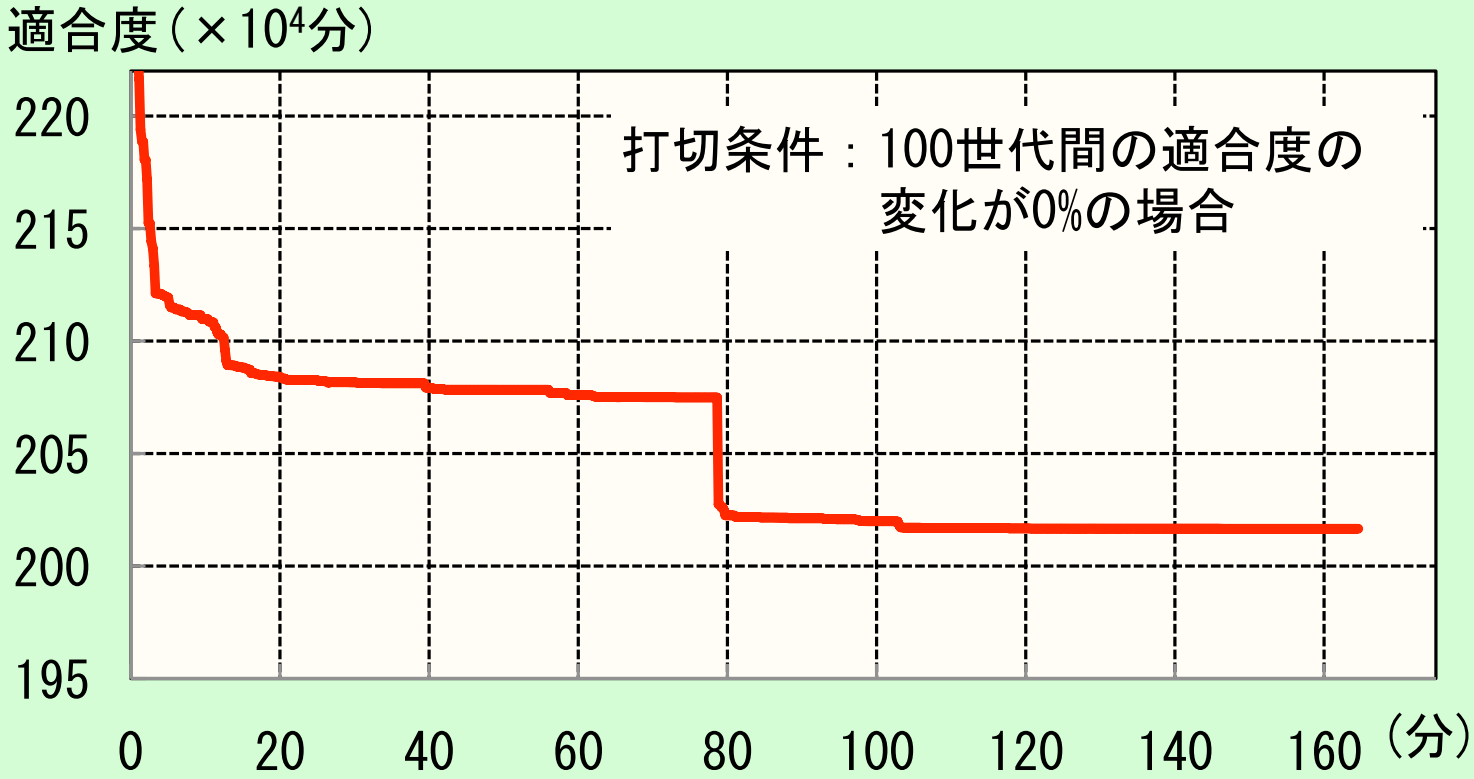


図-10　打切り条件と計算効率

④ 打切り条件の検討 (どこまで計算するか)



図−11　計算の進行状況の例

⑤ 試行回数の検討 (何回繰り返せば、いい結果が得られるか)

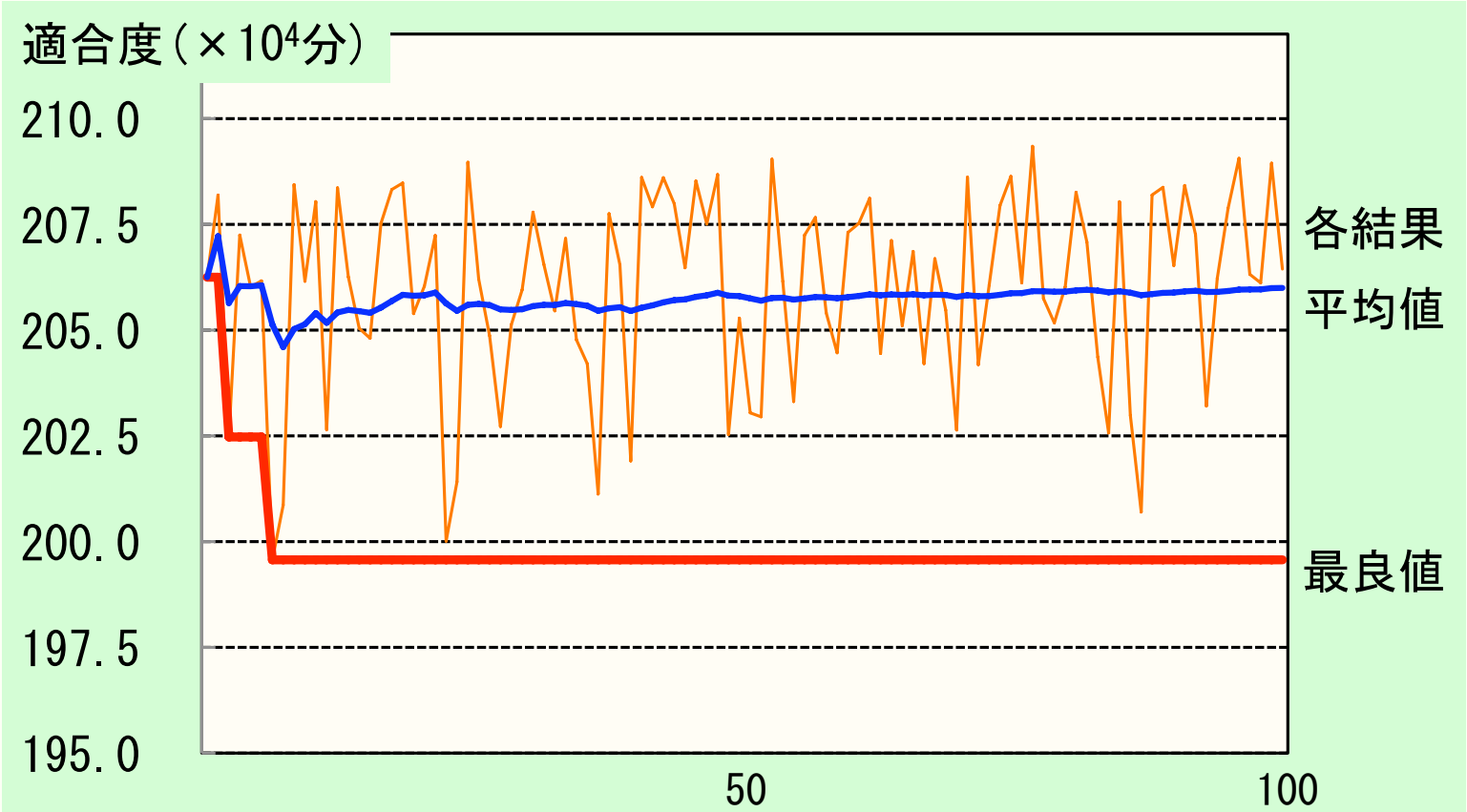


図-12　試行回数と適合度

⑥ 「早熟な収束」対策 (積極的な活用)

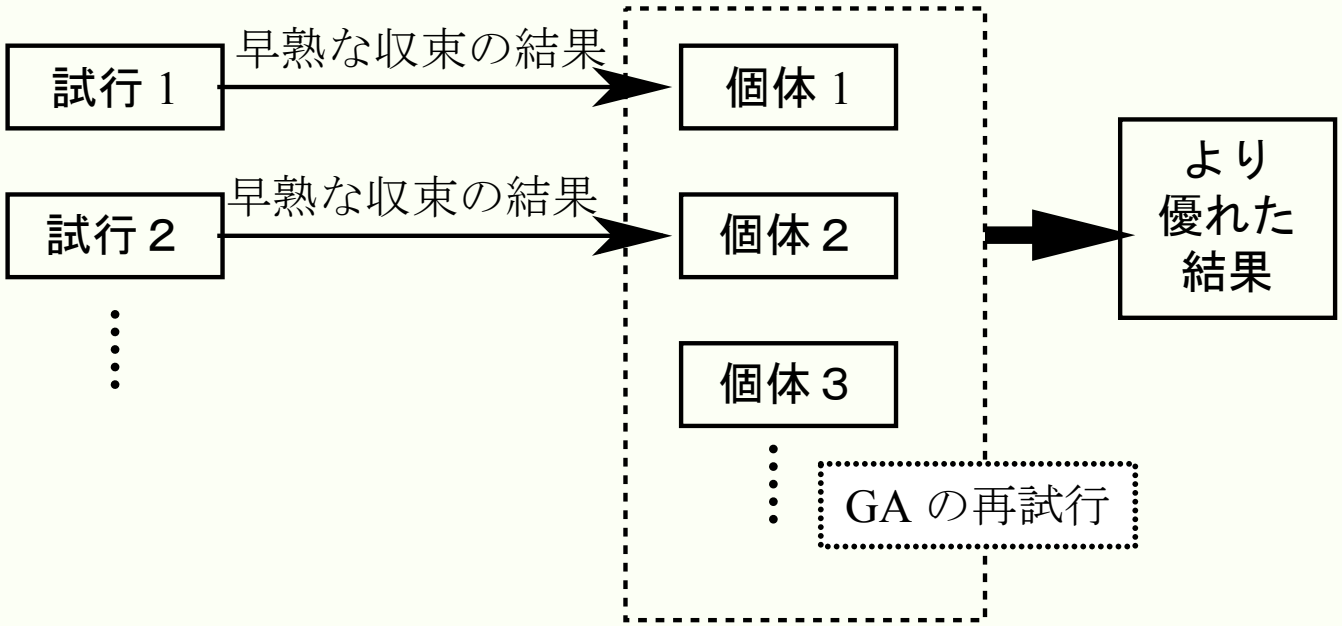


図 － 4　早熟な収束の結果の積極的利用

⑥ 「早熟な収束」対策 (積極的な活用)

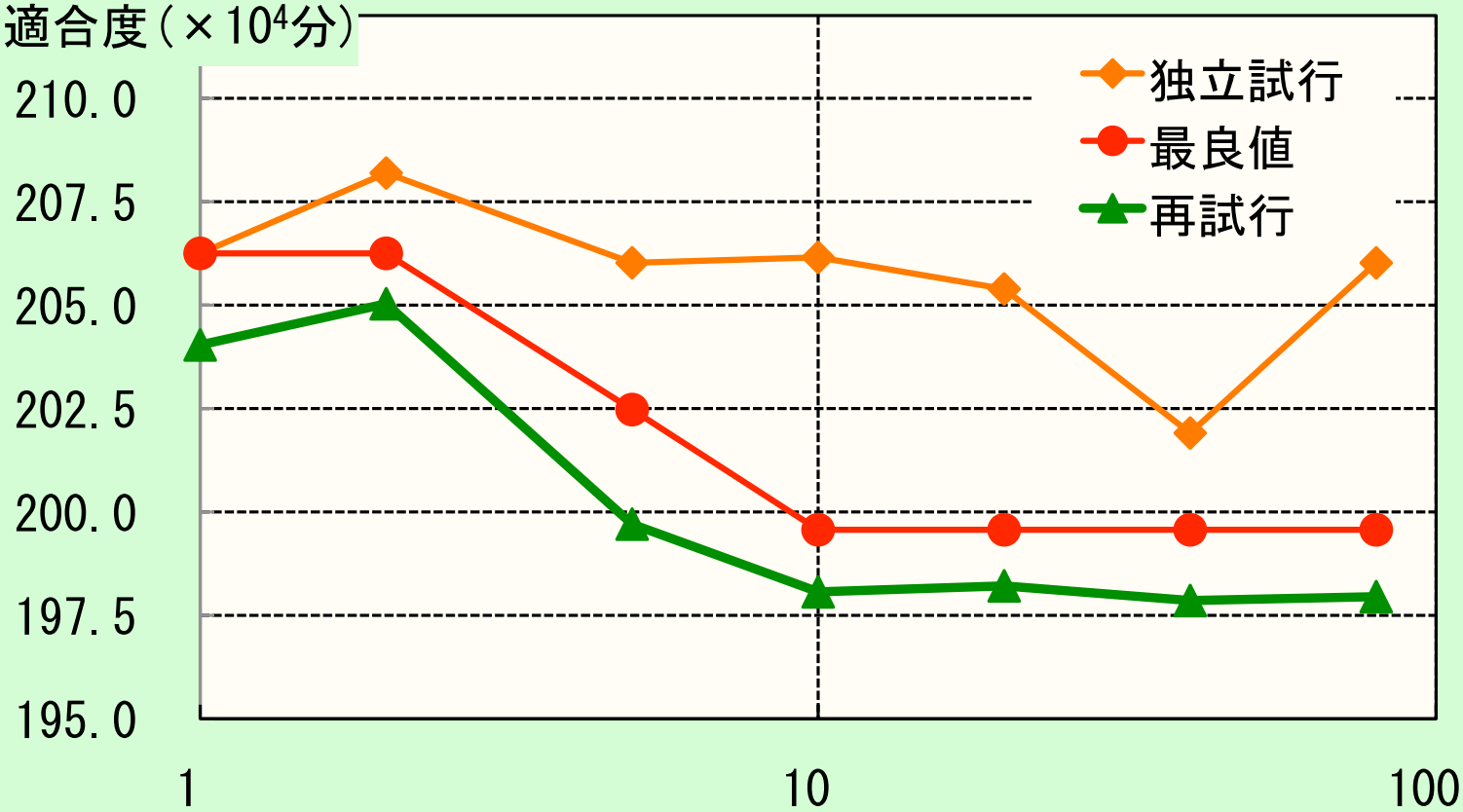


図-13　早熟な収束結果の利用

システムの改良 (まとめ)

① 評価値計算の並列処理

効果有り　ただし、だんだん効率低下

汎用性： CPU以外の部分がボトルネックになる可能性有り

② 集団サイズ (評価対象の個体数) の最適化

効果有り　動的に制御(ビット数)

汎用性： 本研究の分析対象に依存する点は特に無い

③ パラメタ(交叉率,突然変異率)の検討

従来から使用している値でOK

汎用性： 入門書の記載内容の再確認をしたに過ぎない

システムの改良 (まとめ)

④ 打切り条件の検討 (どこまで計算するか)

あまり厳しくしても 大幅な改善無し

汎用性： 適合度計算に 時間がかかる場合は 考慮の余地有り

⑤ 試行回数の検討 (何回繰り返せば、いい結果が得られるか)

あまり多くしても 大幅な改善無し

汎用性： 適合度計算に 時間がかかる場合は 考慮の余地有り

⑥ 「早熟な収束」対策 (積極的な活用)

効果有り　スキーマを効果的に利用

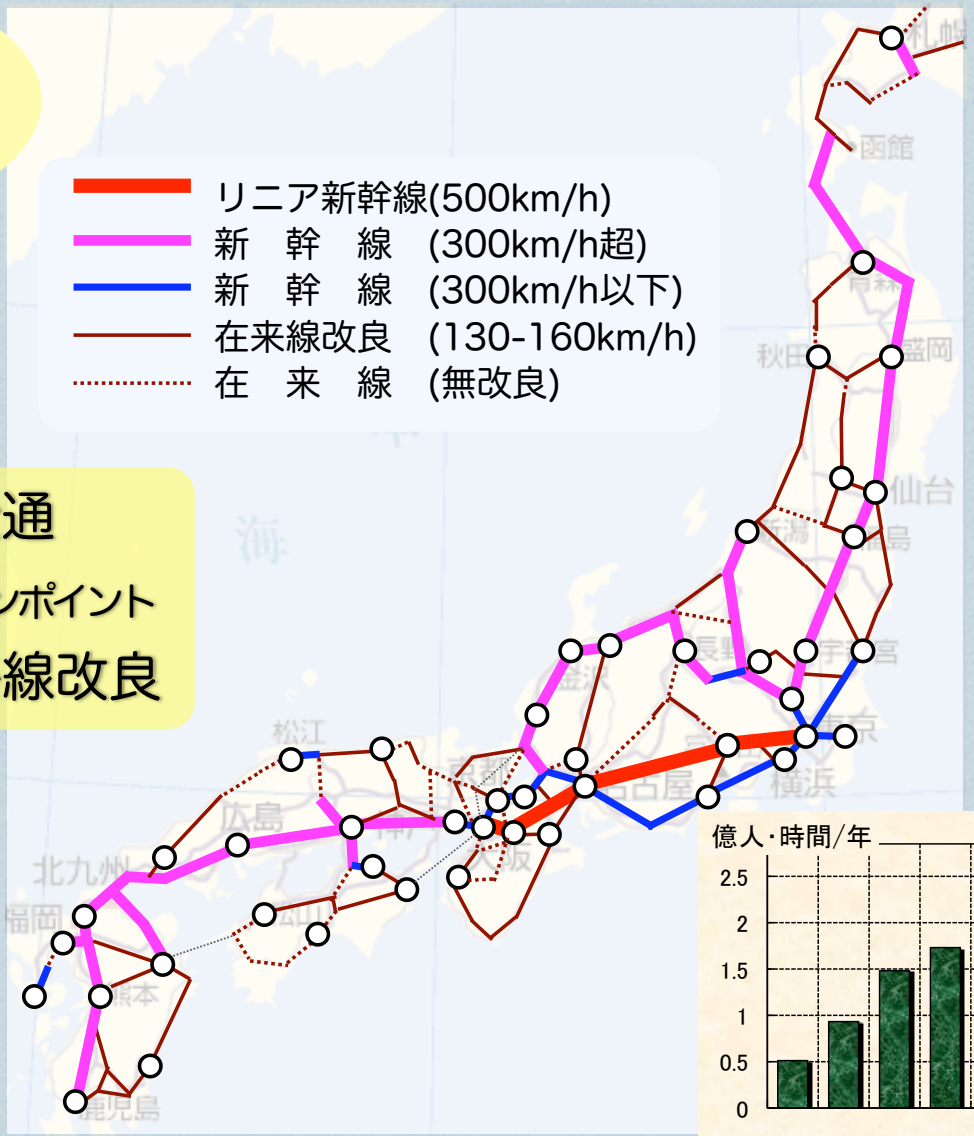
汎用性： 結果が 大きく改善される可能性があり 試す余地有り

効率化の 結果概要

	旧	新	計算効率
CPU	Core2 Duo 2.1GHz	Core2 Quad 4.0GHz (OverClock)	約2倍
並列化	1 core	4 core	約3倍
集団サイズ	初期は1000 段階的に100	100～400程度	約2倍
交叉率		現行条件で OK	1倍
突然変異率		現行条件で OK	1倍
打ち切り条件	100世代 変化率0%	100世代 変化率0.1%	約2倍
その他		コード見直し	約2倍

だいたい
50倍速くらい

15兆円



・リニア：全通
・新幹線：ピンポイント
・広範囲な路線改良

！
所要
時間の
観点で分析

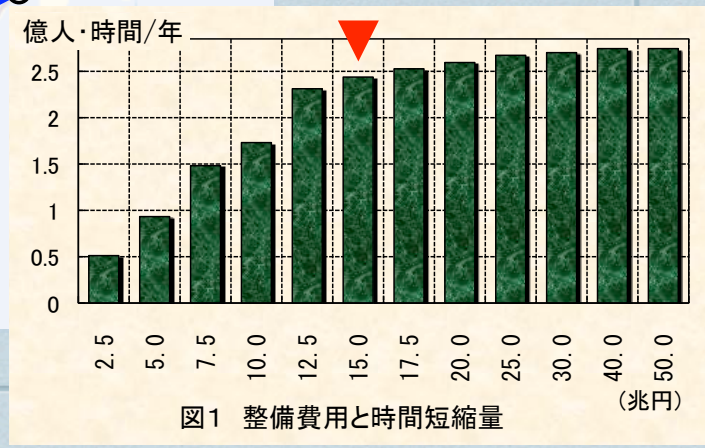


図1　整備費用と時間短縮量

ＧＡを用いた
都市間鉄道網計画策定支援システムの
計算効率化に関する研究

波床　正敏

（大阪産業大学 ）

おわり