

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

A Study on Calculation Efficiency of Support System for Intercity Rail Network Planning with Genetic Algorithm

波床 正敏¹
Masatoshi HATOKO

抄録： これまで新幹線建設と在来幹線の改良に対してどの程度の費用を投入することによってどれだけの利便性向上が得られるかについて、GAを使った数値解析を行ってきた。だが、分析対象が大規模になるにつれて計算時間の長さが障害となりつつあった。分析の結果、並列処理により処理時間は短縮されるが、コア数の増加策には限界があることがわかった。集団サイズは、選択枝の使用状況に合わせて動的に変更する方法が効果的であり、交叉率と突然変異率については適切な値があることがわかった。計算打ち切り条件を厳しくしたり試行回数を増やしても改善は進まないが、試行結果を利用した再試行には大きな効果があることがわかった。

Abstract: A numerical analysis with genetic algorithm has solved a problem of relationship between investment for Shinkansen construction or existing railway improvement and its impact on traveling time with Japanese trunk rails. But the length of calculation time with GA itself has been a difficulty for the study as dealing a larger network. As a result of this study, it has been revealed that increasing CPU-core plan would reach a deadlock in spite of decreasing computer-running time by parallel processing. And it has been also unraveled that GA group size should be dynamically controlled for efficient calculation by information quantity of the group and that cross-over rate or mutation rate ought to be set to a certain number. In addition, it has been found that while strict closure condition or hundreds of calculation trial cannot improve a result, re-trial with some results of trials could dramatically change the goodness indicator into better.

キーワード： GA, 鉄道網計画, 期待所要時間, 最適化問題

Keywords : GA, Railway Network Planning, EVTT, Optimization

1. はじめに

(1) 研究の背景

わが国の全国的な幹線鉄道網計画は、高度経済成長期制定の全国新幹線鉄道整備法(1970)に基づいている。しかし、制定後すでに40年近くの歳月が経過しており、近年では整備新幹線の建設も進行してきているだけでなく、基本計画線のリニア新幹線建設の可能性も見えてきており、在来線改良も含めた国土全体の幹線鉄道網の将来像を再度示すことが必要な時期に来ているのではないかと考えられる。

世界に目を移すと、欧州などで高速鉄道網拡大の傾向にあるが、その中でも特徴的な幹線鉄道政策を実施しているのがスイスである。スイスのRail2000政策¹⁾⁻⁵⁾では、駅間所要時間、運行頻度、発車のタイミングなどを調節することで、複数の拠点駅での乗継ぎ利便性を向上させている。この結果、高速新線の建設延長はさほど長くないものの、ネットワーク全体の利便性を大きく向上させている。このような政策は、今

後のわが国の幹線鉄道政策の参考になると考えられ、各種の検討が必要と思われる。

(2) 研究の目的

本研究に関連する一連の研究⁶⁾⁻⁸⁾では、幹線鉄道網を分析対象とし、新幹線建設と在来幹線の改良に対してどの程度の費用を投入することによってどれだけの利便性向上が得られるかについて、遺伝的アルゴリズム(Genetic Algorithm: GA)を使った数値解析が行われている。

都市圏の通勤鉄道のようなプロジェクトを評価する場合は、プロジェクトの有無比較により効果計測が可能である。しかし、幹線鉄道網のように他路線と連携して機能するような場合には、個々の建設・改良プロジェクトだけを取り上げて効果計測をすることが難しい。このため、プロジェクトの組合せを多数検討しなければならぬが、考え得るあらゆる組合せを全て検討することは不可能である。このような場合に、GAを使った数値解析は比較的高速に実用解を得ることができて便利である。ただし、解の最適性を完全に保証す

1 : 正会員, 博士(工), 大阪産業大学 准教授 工学部都市創造工学科

(〒574-8530 大阪府大東市中垣内3-1-1, Tel: 072-875-3001 (ex. 3722), E-mail: hatoko@ce.osaka-sandai.ac.jp)

ることが難しいことが多い手法であるため、プロジェクト実施の合意形成の材料として使用しようとするような場合には、注意を要するものと考えられる。

GA は比較的短時間に実用解を得る方法であるが、総当たりで探索する方法に比べれば高速であるとはいえず、元来、GA は比較的長い計算時間を要する手法である。また、本研究では実際の乗継ぎ利便性を考慮するために評価関数として期待所要時間(Expected Value of Traveling Time: EVTT)を用いているが、この指標は時間軸に沿って複数回にわたって最短経路探索を繰り返さなくてはならないため、計算時間が長くなりがちである。このため、大規模なネットワークを対象とする計算を行った場合には、計算時間が数日から数週間、場合によっては月単位の時間を要する可能性が高く、国土全体の幹線鉄道網に関する分析しようとする場合には、計算時間の長さが分析の障害となるものと考えられる。

そこで、一連の研究で使用している都市間鉄道網計画策定支援システム⁷⁾について、適切な精度を保ちながら、計算結果をいかに短時間内に得るかについて検討することとした。計算を効率化することについては、計算プログラムを実際にコーディングする際に無駄な部分を排除するなどの地道な工夫等に依存する部分も大きいですが、本研究ではGA特有の各種パラメータ設定や計算手順の工夫などについて考察し、計算の効率化を図ることとした。

(3) GA を用いた交通投資分析に関する既存の研究

GA を用いて交通ネットワークを分析した例としては、道路整備順位決定問題を取り扱った田村らの研究⁹⁾を挙げることができる。予算制約下において道路網に道路を新設したり、あるいは既存道路を改良したりすることによる走行時間の短縮量を求め、これを費用換算したものと投入費用とから費用便益比を計算し、この費用便益比を適応関数としてGAの探索を行っている。これは費用制約下における適応関数の最適化問題であるので、一種のナップサック問題である。

同研究では、仮想的な9ノード17リンクのネットワークに対して、60%と75%の2つの交叉率、2つの異なる突然変異の方法、突然変異率を一定値にする方法と動的に変化させる方法(初期収束の可能性がある場合に突然変異率を50%に上げる)を互いに組み合わせ、計16通りに関して比較分析が行われている。分析の結果、突然変異率を動的に変化させる方法の結果が良好となっている。いずれも計算時間は数分程度であり、同研究後半の17ノード41リンク20工事区間のネットワークを取り扱った分析でも数十分程度となっており、計算時間の長短は問題にされていない。

収束条件としては50世代に達した場合、すべての線列(個体)が同一になった場合、適応関数の最良値が20

世代にわたって更新されない場合のいずれかの条件を満たした場合としているが、経験的に設定されており、体系的に設定されたものとはなっていない。

高速道路ネットワークの段階的整備プロセスに関する青山・松中らの研究¹⁰⁾¹¹⁾¹²⁾では、全国規模の幹線道路網を対象としてGAによる分析が行われており、計算時間の短縮のために「過去の探索結果の効率的利用」が行われている¹⁰⁾。だが、具体的にどのような方法によってどの程度の時間短縮が図られたのかは明確にはされていない。終了条件は3万世代の計算が終了した時点としている。なお、この研究で取り扱われたネットワークに関する分析対象の順列数は約 6.0×10^{52} 通りと記されており、本研究に至る一連の研究⁷⁾で取り扱った九州の幹線鉄道網に関する順列の約 1.5×10^{176} に比べると、世代あたりの計算規模はかなり小さいと考えられる。

鉄道計画についてGAを適用した研究としては、村上・松浦・奥村・塚井らの研究¹³⁾がある。同研究では鉄道と航空の間での機関分担を考慮した運行頻度設定モデルが提案されており、鉄道リンクの運行頻度の設定部分においてGAが用いられている。分析対象としてリンク数が多く、各リンクの水準を示す段階も細かいため、計算は多大であると推測されるが、計算に要する時間については触れられていない。組み合わせ数が膨大であるが、個体数が60に設定されており、世代あたりの計算規模は比較的小さいと考えられる。終了条件としては、事実上6万世代の計算が終了した時点としている。

このように、計算効率の改善を明確な目標として分析が行われている研究はないが、その理由としては、計算時間がさほど長くない、研究遂行の障害とはなっていないからではないかと思われる。しかし、本研究が分析しようとしている対象のように、ネットワークが複雑化したり、適合度計算に時間がかかる指標を採用した場合には計算効率の問題が顕在化する可能性が高く、効率化は重要な課題であると考えられる。

2. 計算システムの概要

(1) システムの全体構成について

本研究で使用する計算システムの全体構成を図-1に示した。計算目的としては、一定の費用制約下で、どの幹線鉄道区間にどの程度の路線改良・新線建設等の投資を行えば、路線網全体としての利便性が最大になるかを計算するものである。システムは、路線ごとの整備段階の選択肢の作成部分と、費用制約下において適合度を最適にする選択肢の組み合わせを求める部分から構成されている。図中にa)~d)の記号が記入された項目があるが、これは第3章で説明する計算効率

化の検討項目である。第3章では、この4項目を含む6項目について効率化の検討を行った。

システム前半の選択肢の作成については、路線の建設単価や改良単価などを設定した上で、例えば表-1のような選択肢を作成する。表-1の区間(久大本線の久留米-日田間 47.6km)では、現状の所要時間は48分であるが、中村線(高知県)の高速化事業の建設単価と実施実績を参考にすると、4.3億円を投入することで全47.6kmのうちの44.3kmに路線改良を実施することが可能であり、所要時間を43分にすることができる。同様に、智頭急行(兵庫県, 鳥取県, 岡山県)において

検討された高速化事業の試算事例を参考にすると、投入金額を16.1億円にすれば39.0kmに対して路線改良を実施することで38分にできる、というふうに、所要時間が5分ずつ短縮されるような選択肢を作成する。なお、この例では、区間両端駅での乗換え時間(または停車時間)を2分と設定しているので、所要時間の値は原則として5分の整数倍から2分を減じた値としている。適切な選択肢が作成できない場合、選択肢作成は省略されるが、例えば表-1の区間では、所要時間33分の選択肢は路線改良では実現できず、160km/h運転の高速新線を建設する方法を採用する。その次は

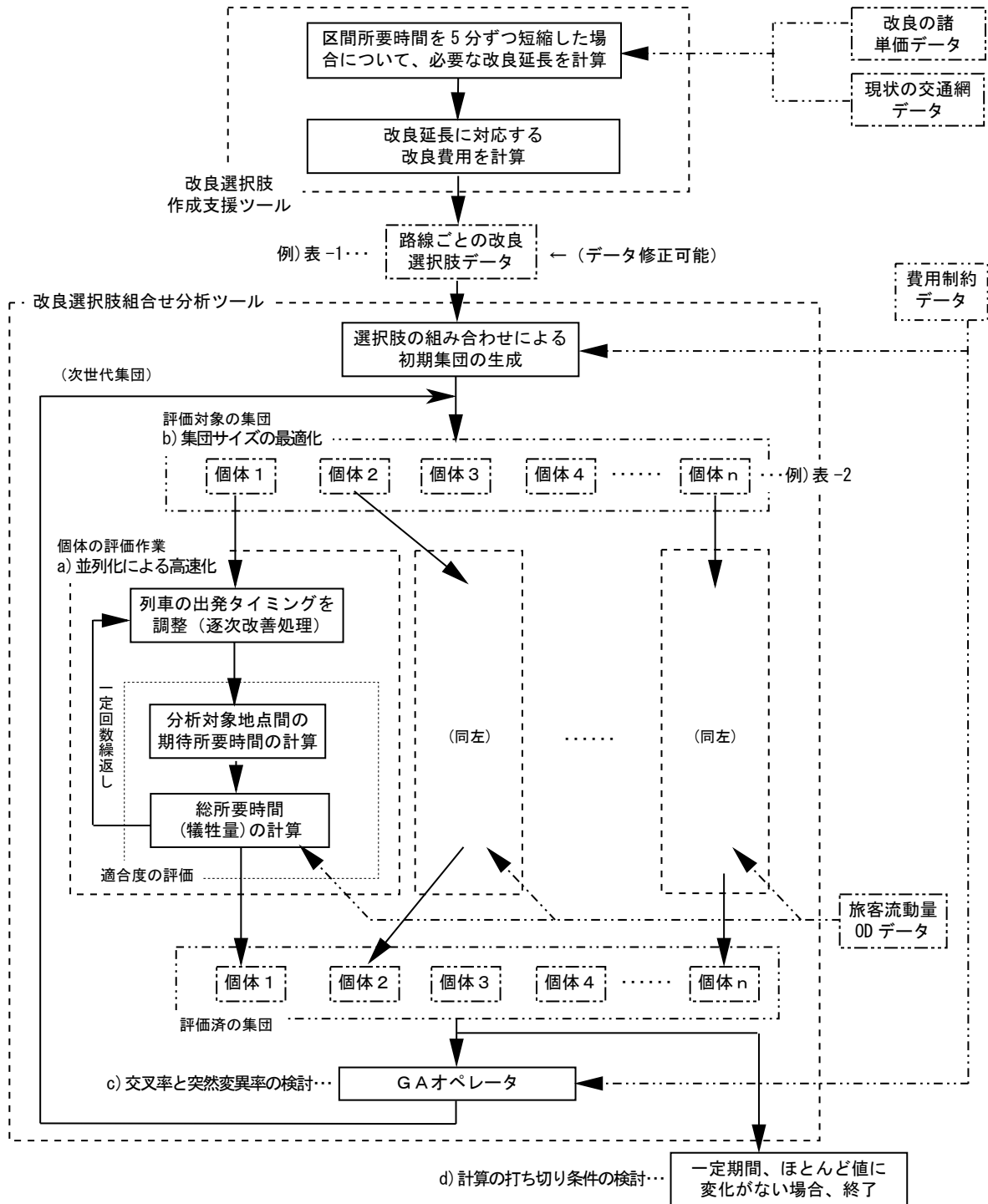


図-1 システムの構成概要

260km/h 運転の高速新線で所要時間が 14 分となる。

システム後半部分については、路線ごと、主要区間ごとに作成された表-1 のような選択肢の組合せについて、列車の出発時刻を各路線・区間・方向ごとに 5 分刻みで変化させた上で、後述の EVTT を用いて適合度(総所要時間)を計算し、最も値が小さくなる選択肢の組合せを GA によって探索する。

表-2 はこのような選択肢の組み合わせの例(一部分)であり、図-1 の「評価対象の集団」の「個体」の内容に相当する。区間ごとの選択肢は表-1 と同様に作成され、その中から 1 つずつ建設・改良内容が選択される(初期集団の場合はランダムに、それ以降は GA オペレータにより作成される)。選択された建設・改良内容に対応する費用の総和が、予め設定した額を超える場合には、選択が再度やり直される。また、適合度の

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

表-2 選択肢組み合わせ(個体の内容)の例

	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
：	：	：	：	：	：	：

値が最小になるように、各区間ごとに順方向および逆方向の列車の出発時刻が調整される。

このような選択肢の組み合わせを表現した個体を多数生成し、それぞれ適合度を計算して評価を行う。本研究ではこの評価部分が並列化可能である。評価を終えた個体は、適合度に応じて交叉や突然変異などの GA オペレーションを経て次世代集団となり、再び評価される。この操作を繰り返すことで、徐々に集団内の個体の適合度が良好になり、最終的に最も良好な適合度の個体を実用解として採用する。

なお、本研究で取り扱う問題は、いわゆるナップサック問題となるが、乗継ぎがうまく行くか否かが適合度に大きな影響を与えるため、単に旅客流動量の多い路線を優先的に改良しさえすれば適合度が良好になるとは限らない。また、乗継ぎの良否には僅かな所要時間やダイヤ設定の差が大きく影響するので、逐次改善のような方法では適切な解にたどり着けないことが予想される。

(2) 期待所要時間(EVTT)について

本システムで適合度を算出する際に用いる期待所要時間(EVTT = Expected Value of Traveling Time)は、次のような考え方に基づく指標である¹⁴⁾。

2 地点間の所要時間は、一般的には利用する便ごとに乗車時間が異なっており、各便の出発時刻においては図-2 の●点のように示される。また、他の時刻を出発時刻とした場合は、次便までの待ち時間が加わり、図中の斜め線のようになる。旅行開始時刻に対して目的地に最も早く着くことを想定すると、斜め線のうち実線部分をつなぎ合わせたノコギリ状のグラフが時刻沿った所要時間の変化を表している。この場合、二点鎖線部分はより早く到着する便があるので利用されない。このノコギリ状のグラフを平均して所要時間相当の指標としたものが EVTT(一点鎖線)である。

EVTT は、各便の所要時間が小さく、運行頻度が高いほど小さな値となり、また各便の所要時間や運行本数が同じ場合でも、団子運転のような実質的な利便性が低い場合には値が大きくなる。複数路線を乗継ぐ場合

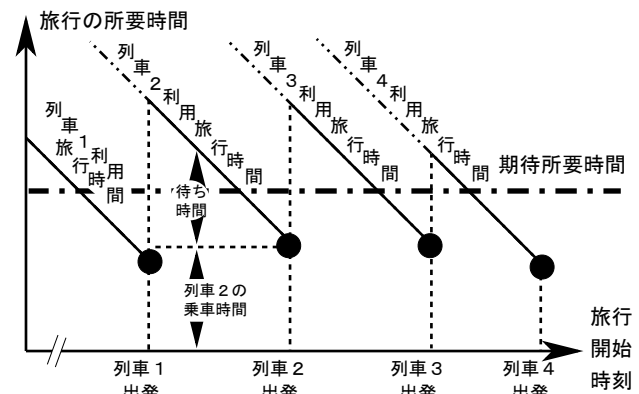


図-2 期待所要時間の考え方

には、実際のダイヤおよび経路に沿って算出することで、乗継ぎの良否も考慮できる。

(3) GAによる探索に長時間を要する原因について

本研究では一定費用制約下における投資額の組合わせという、いわゆるナップサック問題をGAによって解の探索をしており、改良選択枝の組合せだけでも膨大な組合せ数となっており、計算に時間を要する原因となっている。また、各区間を走行する列車についても、出発のタイミングを5分刻みで変化させて計算をおこない、その都度、適合度を計算した上で適合度が最小となる出発タイミングを探索しており、これも計算時間を長くする原因となっている。

さらに、適合度計算において、前述のEVTTを計算しているが、この計算過程は以下の作業の繰り返しであるため、通常の最短経路探索に比べて、出発時刻を多数回設定しなければならない分、長時間を要している。

①出発地点と出発時刻を設定

②列車の出発時刻や乗継げるかどうかを考慮しながら最短経路探索により他地点の到着時刻を計算

加えて、本システムで取り扱っている都市間鉄道における乗継ぎについては、選択した選択枝番号の違いや列車の出発タイミングのわずかな違いによって、乗継ぎの成否の状況が大きく変化することがある。すなわち、適合度が選択枝番号や出発タイミングに対する多峰性関数となっていると考えられ、GAを用いた解の探索を実施したとしても、局所解に陥りやすく、容易に解に至ることができるものではないと考えられる。

3. 計算効率化の検討項目と計算の基本条件

(1) 本研究で検討する計算効率化項目

本研究では、次節で述べるコーディング上の工夫を除く、GA特有の各種パラメータ設定や計算手順の工夫などについて考察することとしている。その主たる項目は以下の各点である。

a) 並列処理による高速化

近年、CPUのマルチコア化が進行しており、安価な小型計算機でも高度な数値処理を行いやすくなってきている。また、GAはいったん初期集団を生成したりGAオペレーションを終えてしまうと、その後は集団内の各個体をそれぞれ独立に評価することが可能であり、並列処理が行いやすい。本システムでは、図-1の中央部の「個体の評価作業」として示した部分を並列処理することで、全体の計算時間の短縮をはかった。並列処理は、MPIやOpenMPなどの技術を使って実現されることが多いが、本研究では比較的簡便な方法で対応した。

b) 集団サイズの最適化

GAによる解析では、集団サイズが大きいほど、より適合度の良好な結果を得る可能性が高まる。しかし、図-1の中央部の「評価対象の集団」に示すように、集団サイズが大きいということは評価対象の個体が多いということでもあり、集団サイズは計算時間に直結する。このため、大きな集団サイズを採用することが難しいことが多い。また、集団サイズを大きくしても、計算の進行に伴って類似の遺伝子を持つ個体が集団内に広まってしまうと、集団サイズを大きくした割には適合度の良好な個体が出現しにくくなるといった事態も考えられる。

本研究では、幹線鉄道改良といった問題の場合、どの程度の集団サイズが適切であるかについて考察を行う。

c) 交叉率と突然変異率の検討

GAによる解析では、交叉率と突然変異率は計算の効率に影響を与える基本的パラメータである。これらパラメータは図-1下部の「GAオペレータ」の部分で使用され、評価済の個体の内容を組み替え、また、変化させ、次の評価対象の集団を生成する際に使用されるため、計算効率に影響を与える。本システムが取り扱う幹線鉄道網については、適合度が選択枝番号や出発タイミングに対して多峰性関数となっている可能性が高く、標準的な設定値では計算効率が必ずしも良くない可能性がある。単峰性関数については、突然変異率を遺伝子長の逆数に設定するなどのセオリーが存在するが、本研究のような一般的な関数については、突然変異に頼る方法では早熟な収束に陥る可能性が高く、かといって交叉に頼る方法ではスキーマを破壊する可能性が高く、交叉率や突然変異率の最適値の決め方は定まっていない。本研究では、どの程度の交叉率や突然変異率が適切かについて具体的に検討を行う。

d) 計算の打ち切り条件の検討

本システムで取り扱う問題に関しては、最適解がどのようなものであるのか事前には明らかになっていない。したがって、何らかの基準を設けて計算を途中で打ち切り、その時点の解を実用解として採用せざるを得ない。打ち切り条件を厳しくすれば、図-1に示した解の探索作業を多数回実行することになり、より適合度の良好な個体を見つけられる可能性が高まるが、計算時間が伸びる原因となる。逆に緩くすれば解の探索作業の実行回数が少なくなり、計算時間は短縮されるものの、良好な個体を得にくくなる。本研究では、どの程度の条件で打ち切るのが適当であるかについて検討を行う。

e) 計算の試行回数の検討

本システムの分析対象のように、分析対象が多峰性関数となっている場合、GAによる探索が進行する

につれて、集団内の多様性が失われると局所解に陥りやすい傾向がある。そのような分析対象については、図-3のように、GAによる計算そのものを複数回試行して最良の結果を採用するという方法もある。この場合、試行回数はどの程度あれば十分であるかについて考察を行う。

f) 早熟な収束の積極的利用

GAによる探索の打ち切り条件を緩くしたり、集団サイズを小さくするなどして短時間で得ることのできる「早熟な収束」の結果については、それ自体を最終的な解とすることは難しい。しかし、そのような結果は部分的に最適な組み合わせ(スキーマ)が形成されている可能性があるため、図-4のように早熟な収束の結果をいくつか集め、これを初期集団(図-1における「選択肢の組み合わせによる初期集団の生成」を実行した直後の評価対象の集団)の一部の個体としてそのまま投入して、GAによる計算を再度実施する。これにより、良質なスキーマが組み合わせられて、より適合度の高い良好な解を得られる可能性がある。本研究では、このような計算手順の可能性について考察を行う。

(2) その他の計算効率化の工夫について

本研究では計算の効率化を目指しているが、本システムは開発当初より計算の効率化に取り組んでおり、局所解対策と計算の高速化などの具体的内容については文献7)に示されている。また、本研究でも文献7)に示された対策や高速化の工夫は基本的にはそのまま引き継がれている。前節(1)に示した検討項目以外については、以下の各点に配慮して計算速度の向上をはかっている。

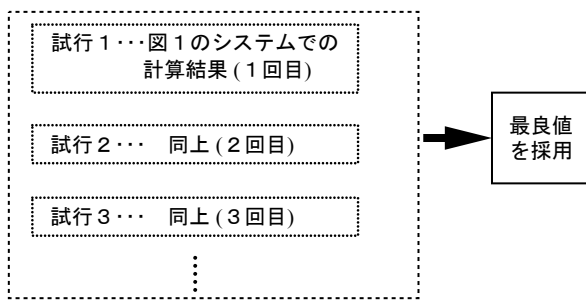


図-3 複数回試行して最良値を採用する方法

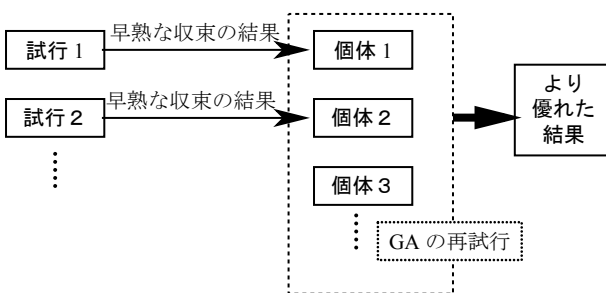


図-4 早熟な収束の結果の積極的利用

①区間ごとに列車の出発タイミングを5分刻みで変更しているが、その都度EVTTを計算(図-1の「分析対象地域間の期待所要時間の計算」の部分)すると計算時間が延びる。そこで、原則として乗継ぎ元の到着時刻直後などに限ってEVTT計算を実施し、それ以外については計算を省略して、EVTT計算の呼び出し頻度を減らした。

②本システムにおいて最も頻繁に呼び出される計算ルーチンはEVTT計算である。このため、EVTTの計算ルーチン外で処理することが可能な部分は全てルーチン外に追い出すなど、実行時間に影響を与える可能性のあるプログラムコードを徹底的に見直した。

③計算時間が週単位や月単位に及ぶため、途中で不慮の停電に遭遇する可能性がある。これに対応すべく、主要データをGAの1世代分の探索が終わるごとにディスクに書き出し、停電発生時には計算を中途から再開できるようにした。

(3) 考察のための計算の基本設定

本研究では、各種の設定等が計算効率に与える影響を考察することを目的としているため、同じ条件でGAによる計算を実施する必要がある(同一実験条件の担保)。そこで、表-3に示す条件に設定した計算機を複数台準備し、各種の条件下で計算を実際に行って考察を行った。

表-3に示した諸元は、標準的な市販品とは異なった設定である。CPUは2種あるが、Q9650(標準ではFSBクロックの9倍で動作)についてはBIOS設定により8.5倍に倍率を下げ動作させ、Q9550(8.5倍で動作)に性能を合わせて使用した。両者とも333MHzが定格FSBクロック数であるが、計算高速化のため、安定動作の範囲内でクロック数を上げている。チップセットは複数種あるが、計算速度には影響しなかった。

OSは64bit版を使用しているが、これはメモリの使用量が大きいからではなく、64bit版コンパイラと組み合わせたときに計算速度が向上するからである。本研究はネットワークを対象として計算しているが、プログラム上でノード間の連結行列のような大規模な配列の使用を避けているので、メモリ使用量は1プロセスあたり最大で約90MB程度である。このため、4章の分析のように4プロセス同時実行した場合であっても、RAMの総容量は2GBあれば十分である(容量を4GB

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

ら終了するまでの計算機の稼働時間そのものであり、コア数が増えるに従って稼働時間が短縮されていることがわかる。しかし、1コアだけで処理した場合は約137分であるのに対し、2コアで処理した場合でも82分を要しており、半分の処理時間(約69分)にはなっていない。また3コアの処理時間は64分、4コアの処理時間は53分であり、それぞれ137分の3分の1である46分、4分の1である34分よりも時間がかかっている。

図-6の黒く塗ったグラフは、計算機の稼働時間にコア数を乗じてCPUコアの稼働時間の累計を計算したものである。計算対象が同じであるため、必要とされる計算量も同じであるはずだが、使用コア数が多いほど稼働時間の累計が増大しており、計算効率の低下が見られる。

(3) 簡易並列処理の課題

本研究における計算プログラムのメモリ使用量は、前章(3)で説明したように比較的小さい。システム全体のメモリ管理方法も特殊な設定は施していない。また、同時に実行している計算プログラム間で相互に通信をしながら計算しているわけでもない。

それにもかかわらず、計算効率が使用コア数増加に伴って低下する原因としては、データが同じ計算機と同じメモリ上に配置されているため、同時にプログラムを稼働させた場合、メモリとCPUの間のデータの転送がネックになっていることが考えられる。

本研究で使用した機材のCPUは二次キャッシュ容量が12MBであるが、二次キャッシュ容量が半分のCPUを使った機材では同じクロック数でも計算時間が大きく伸びる傾向にあることから、上述のデータ転送の問題が大きいのではないかと考えられる。

ディスクアクセスについては、基本的には個体の評価処理の開始時と終了時におこなわれるだけであるので、大きな要因にはなっていないものと思われる。さらに、計算プログラム以外にもバックグラウンドで稼働しているOSの機能が存在しており、これが影響を与えていることも考えられる。

このような影響を少なくするためには、同一計算機内でメモリを共有するのではなく、異なる計算機に評価対象のデータを割り振り、データバスのボトルネックの影響を受けにくくすることも検討課題と考えられる。紙幅の都合で詳細を報告することはできないが、複数の計算機をネットワークで結合して計算した場合、図-6のような頭打ちの傾向を緩和させることが可能であることを確認している。

5. 集団サイズの最適化

(1) 集団サイズに対する考え方

集団サイズについては、サイズが大きいほど、適合

度の良好な個体を見つけられる可能性が高まるが、限度がある可能性がある。

本章では、集団サイズを25個体、50個体、100個体、200個体、400個体、800個体にそれぞれ固定して計算させる方法、および使用されている選択肢数に応じて変化(本研究の分析対象の場合は動的に120~50程度の間で変化)させる方法について比較分析を行う。

選択肢数に応じて変化させる方法とは、分析対象ネットワークで使用されている選択肢数が多いほど、情報量が多く、多様な個体による集団を形成しなければ局所的な解に陥りやすいとの考え方に基づくものである。各区間において使用されている選択肢の数をビット換算し、このビット数を全区間について合計した値を集団サイズとして設定する方法を採用した。GAによる探索が進行すると、選択肢のうち使用されないものが出てくるため、実質的な選択肢数が減少してくる。本システムでは、このような実質的な選択肢数の変化についても考慮し、極力集団サイズを縮小して計算時間の短縮を図った。

(2) 集団サイズに関する計算効率の比較

図-7は前節で述べた集団サイズごとに、GAによる探索を10回試行し、計算に要した時間の平均(以下、平均計算時間)を横軸に、適合度の平均(以下、平均適合度)を縦軸にとって図示したものであり、図中の()内の数値は集団サイズである。また、適合度が最良適合度である試行について、それぞれ計算時間(最良時計算時間)と最良適合度も図示した。

平均値について見ると、集団サイズが大きくなるほど平均適合度が良好になっている。また、最良値についても、ばらつきが見られるものの、集団サイズが大きいくほど最良適合度は良好である傾向にある。しかし、集団サイズが大きくなるにつれて平均、最良ともに計算時間が増大しており、集団サイズが400を超えると計算時間が大きくなるだけで適合度の改善は非常に小さくなる。

図-7には、前節で示した方法で集団サイズを動的に変更して計算した結果(図中に「bit」と表記)も示したが、平均値、最良値ともに集団サイズを固定した方法に比べて短い時間で比較的良好な適合度を得てい

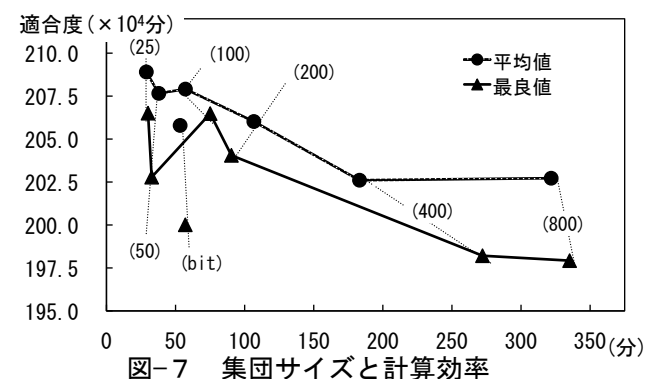


図-7 集団サイズと計算効率

る。これより、選択枝の状況に合わせて集団サイズを変更する方法は計算の効率化に有効であると言える。

(3) 集団サイズに関する課題

本研究では、集団サイズを動的に変更する場合に、使用されている選択枝数をビット数に換算した値をそのまま集団サイズとしたが、ある程度の倍数を乗じて集団サイズとした方がより効率が良い可能性もあり、今後の検討課題である。

6. 交叉率と突然変異率の検討

(1) 交叉率と突然変異率に対する考え方

GAによる解析では、交叉率と突然変異率は基本的なパラメタである。標準的には交叉率が60~95%、突然変異率は5%程度以下が適当とされている¹⁵⁾が、本章では交叉率を50%から90%まで10%きざみで変化させ、適合度と計算時間を調査する。また、突然変異率についても1%、2%、5%、7%、10%の5段階に変化させ、同様に調査する。

(2) 交叉率に関する計算効率の比較

図-8は前節で述べた交叉率ごとに、探索を10回試行し、平均計算時間と平均適合度を図示したものであり、図中の()内の数値は交叉率である。また、最良適合度に関する結果も図示している。

平均計算時間は、交叉率が70~80%の場合に最小であった。適合度は小さいほど良好であるので、交叉率70%を設定値とするのが適していると判断できる。最良値に関しては、交叉率が小さいほど適合度が良好である傾向にあるものの、明確な差にはなっていない。

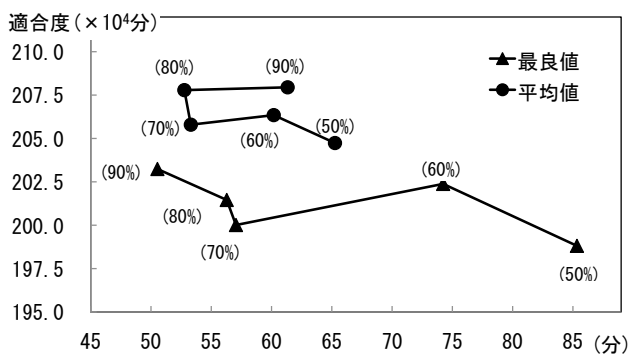


図-8 交叉率と計算効率

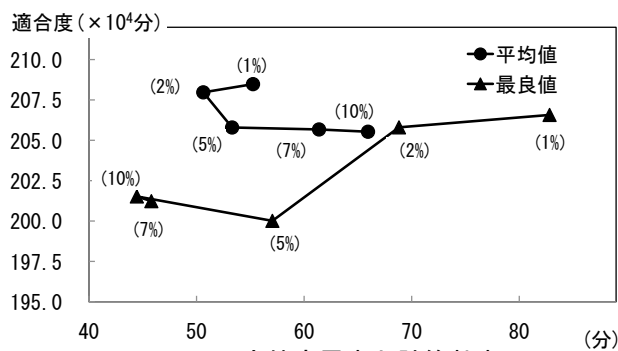


図-9 突然変異率と計算効率

以上より、交叉率を70%程度に設定すると、比較的効率的に良好な結果を得られると言える。

(3) 突然変異率に関する計算効率の比較

図-9は突然変異率を変化させた場合の結果である。平均計算時間は、突然変異率が2~5%の場合に最小であり、平均適合度は突然変異率が高いほど良好な傾向にある。だが、突然変異率5%の場合と10%の場合との差は僅少であるため、突然変異率5%が設定値としては適当であると思われる。最良値についても突然変異率が高いほど良好である傾向にあるものの、突然変異率が5%より大きいと適合度に大差はない。

(4) 交叉率と突然変異率に関する課題

概ね、交叉率70%、突然変異率5%の場合に短時間で比較的良好的な結果を得やすいとの結論になるが、特に突然変異率については2~5%の間でより詳細に検討することも考えられる。また、突然変異率と交叉率の組合せにより、どの程度効率が良くなるかを検討することも考えられる。

7. 計算の打ち切り条件の検討

(1) 計算の打ち切り条件に対する考え方

本システムで取り扱う問題は、事前に適合度の理論的最良値が不明であるため、適合度の値が一定値に達することを以て計算の終了条件とすることが難しい。したがって、適合度の変化が一定期間変化しないことなどを終了条件とせざるを得ない。しかし、どの程度の基準で打切るべきかについては明確な基準が無く、具体的に検討する必要がある。

本システムでは、GAによる探索の過程において、100世代あたりの適合度の変化率が一定値未満になった場合に実用的な解に到達したと判断し、計算を打ち切る方法を採用した。変化率の設定値(以下、打ち切り条件)としては、100世代あたりの変化率が1%、0.5%、0.2%、0.1%、0.05%、0.02%、0.01%、0%(変化無し)の8段階とした。

(2) 計算の打ち切り条件に関する計算効率の比較

図-10は前節で述べた打ち切り条件ごとに、探索を10回試行し、平均計算時間と平均適合度を図示したものであり、図中の()内の数値は打ち切り条件の値である。また、最良適合度に関する結果も同図に示している。

平均計算時間は打ち切り条件を厳しくするほど長くなっており、それに伴って平均適合度も改善の傾向にある。しかし、その差は小さく、時間をかけて計算する意義は小さい。最良値についても最良適合度の差は小さく、緩やかではあるが、打ち切り条件0.1%の場合において最良適合度が最も良好になっている。

これより、打ち切り条件は0.1%程度に設定するのが適当であると結論づけられる。

(3) 計算の打ち切り条件に関する課題

図-11は打ち切り条件0%(つまり、100世代にわたって適合度の改善が全くない場合に終了する)の場合における適合度の変化を、時間軸に沿って示した例である。基本的には探索初期には適合度が大幅に改善するが、探索が進行するにつれて適合度の改善が緩慢になっている。この例では、適合度の改善がいったん緩慢になった後、80分付近で適合度が大きく変化しており、打ち切り条件が緩やかであった場合にはこの変化を待たずして探索を終了してしまう可能性がある。しかし、このような変化がいつ発生するかを事前に予測することは不可能であるため、打ち切り条件を厳しくすることによって適合度の改善を図るような受け身の方法ではなく、もっと積極的に良好な適合度を得る方法について検討すべきであると考えられる。

8. 計算の試行回数の検討

(1) 計算の試行回数に対する考え方

本研究の分析対象のように、GAで多峰性関数を取り扱うと局所解に陥りやすいが、そのような分析対象については、計算そのものを複数回試行し、その中から最良の結果を採用するという方法¹⁵⁾もある。本章では、そのような方法をとる場合、試行回数ほどの程度あれば十分であるかについて考察を行う。検討する試行回数は、最大100回とした。

(2) 計算の試行回数に関する計算効率の比較

図-12は、第3章第3節で述べた基本条件に沿ったGAによる探索を100回試行した結果である。

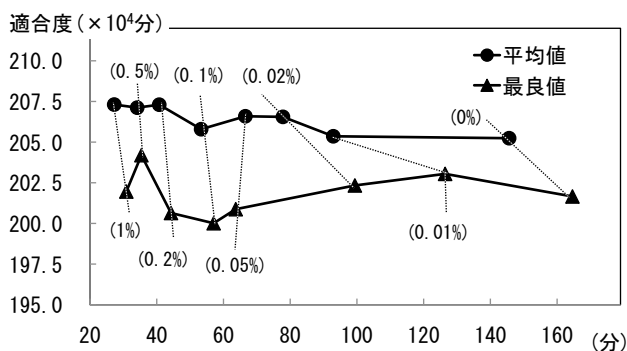


図-10 打ち切り条件と計算効率

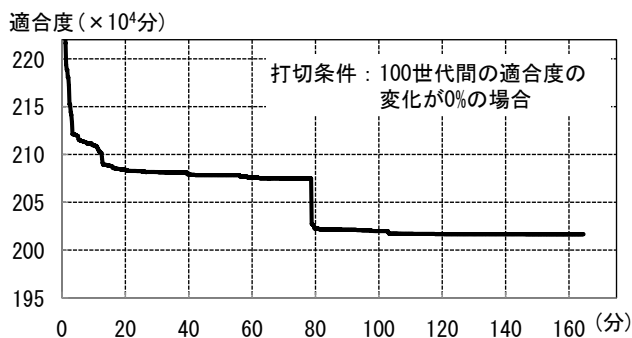


図-11 計算の進行状況の例

同図において、細線で示された「各結果」が試行結果そのままの結果である。100回の試行は互いに独立しており、直前あるいは直後の試行とは何の関連もない。中央付近の中太の線は、その段階までの試行結果の単純な平均の推移である。また、太線はその段階までの試行結果の最良値である。

図-12では、第7回目の試行において100回の試行全体での最良値がすでに出現しているが、その後も20回程度ごとに比較的良好な結果が出現し、試行回数を目安となると考えられる。

(3) 計算の試行回数に関する課題

試行回数についても、前章の打ち切り条件と同様の課題が存在している。すなわち、試行回数を増やせば良好な回を得られる可能性は向上するものの、そのような良好な結果となる試行がいつ生じるかを事前に予測できない。このため、試行回数を増やして適合度の改善を図る方法ではなく、もっと積極的に良好な適合度を得る方法を検討すべきであると考えられる。

9. 早熟な収束の積極的利用

(1) 早熟な収束の積極的利用に対する考え方

GAにおいて局所的な解に陥ることを「早熟な収束」と言うが、本研究の分析対象の場合、計算時間を短縮しようとする、このような早熟な収束に至る可能性が高くなる。しかし、このような早熟な収束に至った場合でも、部分的に乘継ぎ状況が最適化された状態(つまり、良質なスキーマが形成されている状態)である可能性が高い。そこで、GAによる探索を再度実施し、その際に図-4のように集団の一部に既に得られた探索結果を混入させることで、より適合度の高い良好な解を比較的短時間に得られる可能性がある。

(2) 早熟な収束の積極的利用の計算効率

図-13は、GAによる探索(以下、再試行)の際に、前節の方法で図-12の探索結果(以下、独立試行)を評価対象の集団に加えた結果を示したものである。横軸は独立試行時の試行回数(わかりやすくするため、対数目盛にしてある)、縦軸は適合度である。「独立試行」は図-11の「各結果」と同一であり、「最良値」は図-11

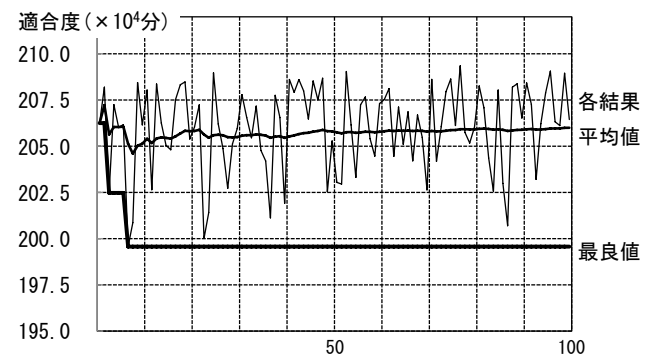


図-12 試行回数と適合度

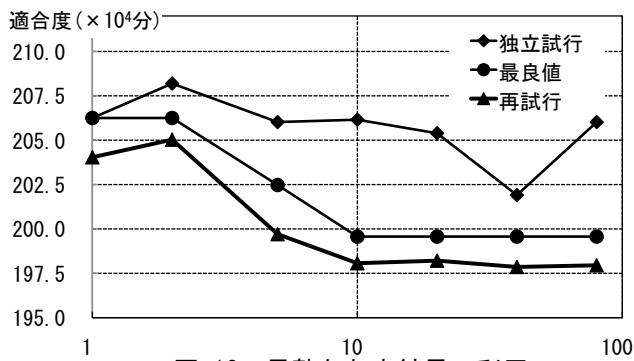


図-13 早熟な収束結果の利用

の「最良値」と同じである。第1回から第100回までのすべての試行について再試行を実施することは煩雑なので、第1, 2, 5, 10, 20, 40, 80回の各独立試行に関してのみ再試行を実施した。図-13では、例えば、第10回に対応する「再試行」については、まず「独立試行」の第1回から第10回までの計10個の探索結果を準備し、GAによる探索の際にエリート保存戦略を採用する場合と同様に個体集団に混入させた。なお、再試行は1度だけ実行した。

図-13によると、結果を再利用して1度だけ再試行しただけであるが、適合度が大幅に改善されており、例えば5回の独立試行結果を再利用して得た結果は、100回の独立試行による最良値とほぼ同じであった。すなわち、計算の効率が10倍以上になったと言える。また、独立試行時の実行時間は1回あたり60分弱程度であるが、再試行の実行時間は30分程度であり、この面でも効率的である。

(3) 早熟な収束の積極的利用に関する課題

以上のように、独立試行の結果を再利用してGAの探索をおこなうことは、計算の効率化にきわめて有効であることがわかった。図-13に示した分析では、再試行回数が1回であるが、これを複数回に増やして最良値を採用するなどの方法をとることで、さらなる結果の改善が期待される。

10. 研究のまとめと今後の展望

(1) 本研究のまとめ

本研究の結果をまとめるとともに、結果の汎用性などについて考察すると、次のようになる。

a) 並列処理について

まず、簡易並列処理を導入した結果、処理時間は短縮された。しかし、CPUのコアあたりの処理能力は低下し、CPUのコアを増やし続ける方法では計算効率の改善に限界があることがわかった。

一般的にGAでは、適合度の計算対象となる集団をひとたび生成してしまえば、その後は適合度(評価値)の計算は個体ごとに行えばよいので、この部分を並列処

理しやすい。本研究でも個体の適合度計算部分を並列化した。本研究で取り扱った例のように適合度の計算方法が複雑(EVTTの計算を経てその総和を適合度とする)で、メモリ上のデータへのアクセスが多い場合には、単一の計算機上で並列処理を行うと効率低下につながりやすいと言える。しかし一方で、適合度の計算処理時間が大きいことは、計算処理を行っているプロセス間の通信等のオーバーヘッドの割合を相対的に低下させることになる。すなわち、通信ネットワーク上に配置された複数の計算機で並列処理を行う方が効率が向上する可能性が高くなる可能性がある。

逆に適合度計算が比較的簡単でメモリ上のデータへのアクセスが少ない場合には、単一の計算機上で並列処理を行った方が有利と思われる。

b) 集団サイズについて

集団サイズについては、サイズを固定して探索を行った場合、集団サイズが400を超えると結果が改善されにくくなることがわかった。集団サイズを選択肢の使用状況に合わせて動的に変更する方法を採用すると、比較的短時間に良好な結果を得ることができるとわかった。

一般的にGAでは解の探索の進行に伴い、集団内の多様性が失われ、似た遺伝子を持つ個体が増える。多様性に関しては、遺伝子間のハミング距離などを使って多様性指標を計算し、淘汰圧の調整などに用いられることが多いが、これは細部にわたる計算精度を高める反面、計算時間を増大させる可能性が高い。

本研究では選択肢の使用状況という一種の多様性指標を集団サイズの調整に用いており、一定の計算精度を保ちながら時間短縮することができた。このような手法は一般的なGAにおいても有効と考えられる。

c) 交叉率と突然変異率について

交叉率と突然変異率について検討した結果、本研究の対象とする問題については、交叉率が70%程度、突然変異率については5%程度に設定するのが適切であることがわかった。

このような交叉率や突然変異率の値そのものについては、一般性はないと思われる。しかし、一般的なGAの探索においても、計算時間や最終的な適合度(計算精度)は交叉率や突然変異率の設定値によって少なからず変化することが予想される。適合度計算が複雑で計算時間が長くなることが予想される場合には、類似の小規模な問題を検討し、適切な交叉率や突然変異率をあらかじめ求めた上で本来の問題に取りかかることで、計算効率が向上することが期待される。

d) 計算の打ち切り条件について

計算の打ち切り条件については、条件を厳しくしても結果が大きくは改善されないことがわかった。具体的には、100世代あたりの適合度の変化率が0.1%程度未

満の場合に終了するように設定するのが適切であることがわかった。

この打ち切り条件の具体的な値そのものについても一般性はないと思われる。しかし、計算効率を考慮して打ち切り条件を設けることには意義がある。GAを用いた研究では、何万世代にもわたってほとんど適合度が改善されないことを確認してから計算を打ち切っている例¹⁰⁾¹³⁾を見かけるが、本研究のような適合度計算に長い時間を要するような分析対象の場合には、このような方法は事実上採用できない。

そこで、本研究のように一定の基準を設けていったん計算を打ち切り、後述の再試行のような方法で精度を高める方が、小さな確率で生じるかもしれない適合度改善を延々と待つよりも有用であると考えられる。

e) 探索の試行回数について

GAによる探索を複数回試行する方法については、大幅に試行回数を増やしても、それに応じて最良値が明確に改善されるわけではないことがわかった。

GAによる探索を複数回試行する方法は、局所解に陥りやすい問題を取り扱う場合において有効な、汎用的手法であると考えられる。しかし、回数を増やせば増やすほど結果が改善するわけではなく、限界がある。

f) 早熟な収束の積極的利用について

試行回数を単純に増やすのではなく、試行結果を利用して再試行することによって結果が大幅に改善されることがわかった。

早熟な収束の結果(局所解)が良質なスキーマを含んでいる可能性があることを利用した再試行法は、汎用的に利用可能と思われる。一般的には、局所解を避けるには淘汰圧を適切にコントロールしながら探索を進めるが、淘汰圧の調整度合いは繰返し計算を行って経験的に得るものであるため、実際には適切にコントロールすることが難しい。本研究のような方法はそのような経験的な要素が無くても利用可能と思われる。

(2) 本研究の適用例と今後の展望

本研究では、九州を分析対象として計算効率改善に関する各種の考察を実施し、いくつかの重要な知見を得た。一連の研究の大きな目的は、よりネットワーク規模が大きい全国の幹線鉄道網に関する分析を実施することである。

本研究で得られた手法は、全国規模の幹線鉄道網を対象とした最近の研究¹⁶⁾に適用されている。同研究では、全国規模の鉄道網を対象とするとともに、計算の諸条件を多数設けて多数回の分析を行っており、本研究による効率化無くしては、事実上成立しない。

今後は、効率化された計算システムを発展させ、より高度で複雑な評価基準(適合度計算)を導入した分析なども実施できるものと考えられる。

参考文献

- 1) SBB: Information - Bahn 2000, German, <http://mct.sbb.ch/mct/infrastruktur/infrastruktur_bahnbetrieb/bahn2000.htm>, (入手 2009. 5. 26).
- 2) SBB: Rail 2000 - A Public Transport Network for the Third Millennium, <<http://mct.sbb.ch/mct/en/bahn2000-summary.pdf>>, (入手 2009. 5. 26).
- 3) SBB: Bahn 2000-Grundlagen-Knotensystem. <http://mct.sbb.ch/mct/en/infrastruktur/infrastruktur_bahnbetrieb/bahn2000/bahn2000-grundlagen/bahn2000-grundlagen-knotensystem.htm>, (入手 2009. 5. 26).
- 4) 波床正敏・中川大: 幹線鉄道におけるハブシステム構築の効果と意義に関する研究 - スイスの鉄道政策 Rail2000 の効果分析を踏まえて -, 都市計画論文集 No. 41-3, pp.839-844, 2006.
- 5) HATOKO Masatoshi, NAKAGAWA Dai: Comparative Analysis of Swiss and Japanese Trunk Railway Network Structures, 11th World Conference on Transport Research, WCTRS, 2007.
- 6) 波床正敏・中川大: 公共交通網におけるパルスタイムテーブルシステム成立条件に関する研究, 土木計画学研究論文集 Vol.24, no.4, pp.693-702, 2007.
- 7) 波床正敏・中川大: GAを用いた都市間鉄道網計画策定支援システムの開発, 2007年度土木情報利用技術論文集 Vol.16, pp.71-82, 2007.
- 8) 波床正敏・中川大: 幹線鉄道整備の基本方針がネットワーク形成に与える影響に関する研究, 土木計画学研究論文集 Vol.25, no.2, pp.487-498, 2008.
- 9) 田村亨・杉本博之・上前孝之: 遺伝的アルゴリズムの道路整備順位決定問題への適用, 土木学会論文集 No. 482/IV-22, pp. 37-46, 1994.
- 10) 青山吉隆・松中亮治・野村友哉: 「大規模高速道路ネットワークの段階的整備プロセスの最適手法とその応用」運輸政策研究 Vol. 5, No. 2, pp. 2-13, 2002.
- 11) 松中亮治・柚木俊郎・青山吉隆・中川大: 「わが国における高速道路ネットワークの段階的整備プロセスの事後評価」, 土木計画学研究・論文集 Vol. 20, No. 1, pp. 33-42, 2003.
- 12) 松中亮治・谷口守・青山吉隆・舛岡田渡史: 「高規格幹線道路網整備計画における段階的整備プロセス」, 土木学会論文集 No. 793/IV-68, pp. 13-25, 2005.
- 13) 村上直樹・竹内太郎・奥村誠・塚井誠人: 「航空との補完的サービスを考慮した最適鉄道運行計画」土木計画学研究論文集 Vol. 23, no. 3, pp. 629-634, 2006.
- 14) 天野光三・中川大・加藤義彦・波床正敏: 都市間交通における所要時間の概念に関する基礎的研究, 土木計画学研究論文集 9, pp.69-76, 1991.
- 15) 伊庭斉志: 遺伝的アルゴリズムの基礎, オーム社, 1994
- 16) 波床正敏・中川大: GAを用いた幹線鉄道網分析における条件設定の影響に関する研究, 土木計画学研究講演集 39, CD-ROM, 2009.

(2009.5.29 受付)