

遺伝的アルゴリズムを用いた デマンドバススケジューリングに関する基礎的研究

Application of Adaptive Co-evolution Genetic Algorithm to Demand Bus Scheduling

古田 均*, 亀田学広** 加藤強士***

Hitoshi FURUTA, Takahiro KAMEDA and Tsuyoshi KATO

*工博 関西大学教授 総合情報学部（〒569-1095 大阪府高槻市霊仙寺町 2-1-1）

**関西大学大学院 総合情報学研究科 応用ソフトコンピューティング専攻

***関西大学 総合情報学部 総合情報学科

Recently, the number of bus passengers is decreasing due to the distribution of own cars. This causes the reduction of scheduled bus service, which further decreases the number of passengers. Under such situations, demand bus service gains attention as a possible alternative, and actually, the demand bus system has been employed in several local cities. The demand bus system means the bus service that can change the schedule of bus service such as timetable and route according to the demand of users. In order to make the demand bus system effective, ITS technology can be used to satisfy the requirements of passengers and achieve the appropriate scheduling from the standpoint of bus management. In this paper, an attempt is made to develop a decision support system that can provide a reasonable and practical scheduling plan to optimize the timetable and route. It is noted that the system is intended to minimize the necessary cost and the traveling and waiting time of users. Numerical examples are presented to demonstrate the applicability of the system proposed here.

Key Words: genetic algorithm, demand bus system, scheduling, traveling salesman problem

1. 緒 論

現在、わが国では自家用車の普及に伴い路線バスの利用者が減少し、経営の問題から便数が減り、利用者が更に減少する悪循環に陥っている。特に地方の過疎部でこの傾向が顕著に現れている。そこで、路線バス利用者の持つ様々な不満を解決するために ITS (Intelligent Transport Systems: 高度道路交通システム) の技術を用いたデマンドバスシステムが注目されており、一部では実際に運行されている。

デマンドバスとは利用者から受け付けた予約に応じてバスの運行スケジュールを変更、決定するバスシステムである。またデマンドバスシステムでは顧客の満足度を向上させるために、新たにバス停を設け運行ルートを増加させる等の対策を行っている。デマンドバスシステムでは、発生する多くの予約（デマンド）に対して、できるだけ各デマンドを満足させつつ、経営的側面からも問題のない運行スケジュールが必要である。

本研究では、バスの便数と各バスが担当するデマンドおよび巡回路を同時に最適化しつつ、運行コストを抑えたスケジューリングを行うシステムを開発する。スケジュールの決定に改良型遺伝的アルゴリズム¹を適用し、その有用性について検討を加える。

2. デマンドバスシステム

従来の一般路線バスは、幹線道路沿いに場所、時間の双方の観点から、まとまった量の利用者を見込めることから、経路、バス停、ダイヤなどが事前に決定され、利用者は安心してバスを利用し、バス事業者は効率的な運用が可能であった。しかしながら、バス利用者の減少、利用者のニーズの多様化、バス走行環境の悪化などの理由から一般路線バス方式での運行では採算がとれない路線がでてきている。これらの問題を解決するため、利用者の需要に応じて柔軟にバスを運行するデマンドバス方式が注目されている。地域特性に合わせ迂回型、ピックアップ型、完全型などが考えられている。デマンドバスは高齢化社会、都市部における交通弱者、地方過疎化問題、環境・エネルギー問題などの観点からも注目を集めている。

デマンドバスシステムは管理センターで集中管理され、電話、FAX、Web、情報端末などから予約を受け付け、それに基づき運行経路、配車の決定を行う。予約によって利用者の待ち時間を減らし、効率よく運用することでバスの運行空き時間を解消することができる。このような利点を享受するには、デマンドバススケジュールの最適化が重要な鍵となる。

デマンドバススケジュールの最適化を行うには、適切なデマンドの巡回路及とデマンドの割り当ておよびバスの台数を決定する必要がある。

3. 改良型遺伝的アルゴリズム

遺伝的アルゴリズム(以下 GA)とは、生物の進化に着想を得た確率的探索手法の一つである。問題の解候補を個体として GA 空間に写像し以下に示すような交叉、突然変異、自然選択といった遺伝的オペレータを適用する。

- 1) 初期世代発生
- 2) 選択
- 3) 交叉・突然変異
- 4) 淘汰

1)で乱数等により問題の解候補である個体群を生成する。次に終了条件を満たすまで、あるいは指定回数 2)から 4)のオペレータを繰り返し適応する。各個体は環境に対する適応度を持つ。環境とは問題空間を表し、適応度は目的関数に対する評価である。各個体は、染色体すなわち DNA としての側面である遺伝子型(Gene Type : GTYPE)と解候補としての側面である表現型(Phenomenon Type : PTYPE)を持つ。GA オペレータは、遺伝子型に対して行われる。

GA は問題空間中を効率よく探索を進めるために、有用な解候補(個体)のいくつかから、新たな解候補(子孫)を作りだし、解候補の近傍を並列的に探索する。また、初期に発生させた個体の持つ可能性以外にも考慮するために突然変異を行い、探索空間を広げる。

有用な個体から子孫を作り出す GA オペレータは、単純に個体の持つ遺伝子(遺伝子型)を操作するのみであり、特別な知識を用いるわけではない。様々な問題に応用されている汎用性の要因の一つと考えられる。有用な解を判断する評価関数は、遺伝子型からデコーディングした解候補である表現型に対して行われる。さらに対象とする問題空間を GA 空間に写像するルールをコーディングルールと呼ぶ。もちろんコーディングルールには、交叉の方法、突然変異などの GA オペレータの設計も含まれる。

実問題を効率よく解くには、コーディングルールを適切に設計することが肝要である。

本研究では、各バスが担当するデマンドとデマンドの巡回路およびバスの台数を同時に最適化するために、GA を改良し適用する。

4. デマンドバススケジューリングへの適用

4.1. コーディングルール

本研究では、デマンドバスの台数及び各バスが担当するデマンドとデマンドの巡回路を最適化するため Fig.1 に

示すように、デマンド部位と各バスの担当デマンドを決定する割り当て部位から個体の染色体を構成する。

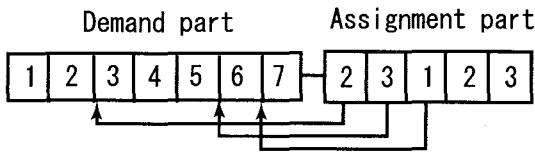


Fig.1 Structure of chromosomes for Demand Bus Scheduling

まず、利用者から受け付けた予約(デマンド)に対して一意に ID をつけ、予約(デマンド)ID を遺伝子として、デマンド部位の染色体を構成する。デマンド部位の遺伝子情報である ID は、重複しないようにランダムに決定する。

割り当て部位の染色体は、デマンド部位を分割する方法を遺伝子情報として持つ。デマンド部位の遺伝子長は、デマンド部位の細分を行うに十分な長さを取り、各バスのデマンド割り当て数をコードする。割り当て部位の情報に従い、デマンド部位を分割する。分割するデマンドがなくなった場合、以後のコードは使用されない(Fig.2 網掛け部)。また表現型であるデマンド巡回路は、バス運行ルートに変換される。各デマンドの乗車するバス停 ID を参照し、これらを結ぶネットワークを作る。バスの巡回経路は割り当てられたデマンド ID から予約時間を参照し、時系列順に並び替え巡回経路ネットワークを生成する。バス停間の経路はあらかじめ決められたものとする。

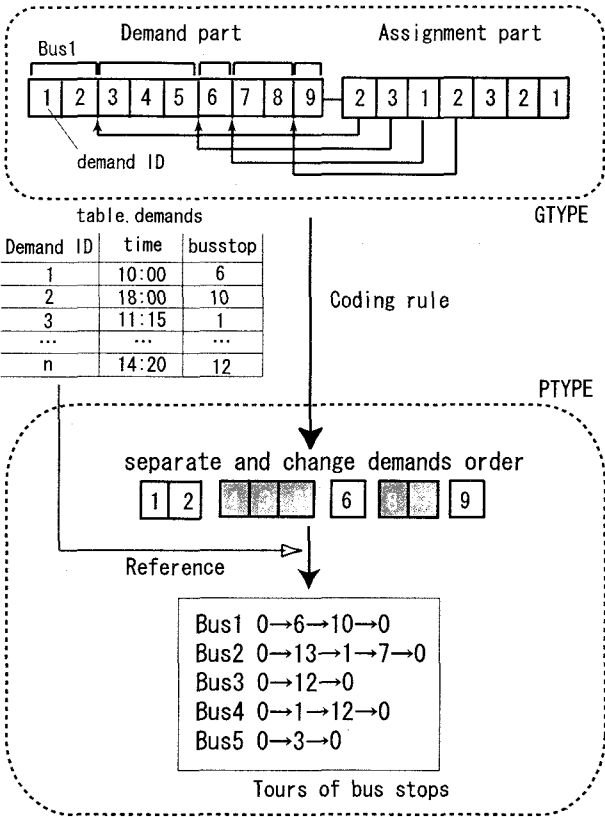


Fig. 2. PTYPE and GTYPE

本研究では、ダイクストラ法により最短距離をもつ経路を選択させている。Fig.2に PTYPE と GTYPE の変換例を示す。

交叉、突然変異オペレータは各種制約条件を満たすように行う。デマンド部位は、任意の位置で切断し両親の染色体を接続する通常の交叉を用いると制約条件を満たさない致死遺伝子を簡単に作り出してしまふ。制約条件は、デマンド ID は重複しないことである。そこでデマンドの順位を保存しつつ、重複が起これないように交叉²を行う。まず任意の交叉点 2 点を選択し、中央部を各子供の染色体に転写する。Fig.3 に示す①のように染色体の先頭部から走査し、①にある遺伝子が、親 B に現れる順序に従い遺伝子を転写する。同様に②のように後尾部から走査し、現れる順序に従い遺伝子を転写する。

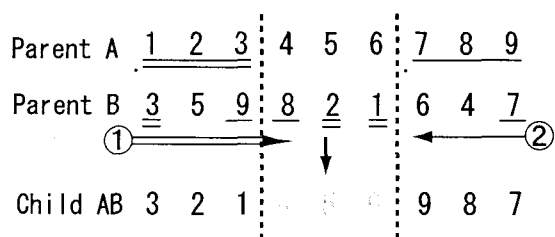


Fig. 3. Crossover at demand part

デマンド部位の突然変異は、ランダムに選択した 2 点を入れ替える方法と、2 点間の遺伝子順序を入れ替える方法を用いる(Fig.4)。

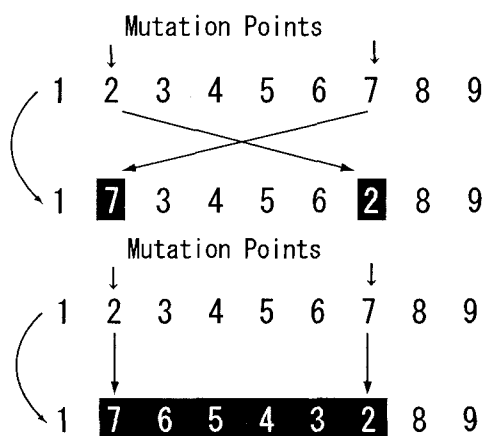


Fig. 4. Mutation at demand part

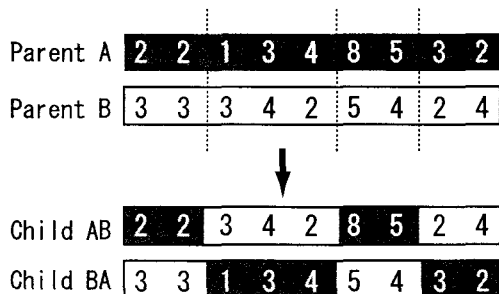


Fig. 5 Crossover at assignment part

一方、割り当て部位は多点交叉を用いる。Fig.5 に示すように任意に選んだ点で染色体を切断し、遺伝子を入れ替え、子の遺伝子に転写する。突然変異は、任意に選んだ遺伝子を乱数によって異なる値を入れる。各遺伝子座には整数値を用いる。

本研究では実用に向け、ユーザーの目的地も考慮に入れたスケジューリングを行うために初期の線列を生成するアルゴリズムに工夫を行った。

ユーザーの乗り降りを考慮した場合、制約条件は、デマンドの巡回は時系列、あるユーザーの降りるバス停より先に乗るバス停を巡回する必要がある。以上の制約条件のもとに発生する組み合わせは Fig.6 に表す構造をとる。

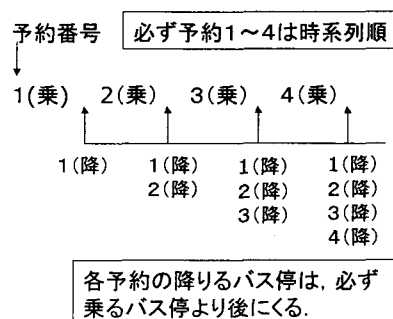


Fig. 6

制約条件を満たす線列の中で、様々な組み合わせが存在することが分かる。本システムで採用する交叉アルゴリズムは初期の線列において制約を満たす限り、制約を破る線列を作り出さない。そこで Fig. 7 に示すように、乗る時間によってソートし、デマンドの乗り場、降り場にそれぞれ偶数、奇数の ID を付ける。

予約ID	乗るバス停	降りバス停
12:30	1	4
13:45	5	4
9:45	3	10
8:45	10	7
14:00	9	1

予約ID	乗るバス停	降りバス停
8:45	10 ①	7 ②
9:45	3 ③	10 ④
12:30	1 ⑤	4 ⑥
13:45	5 ⑦	4 ⑧
14:00	9 ⑨	1 ⑩

Fig. 7

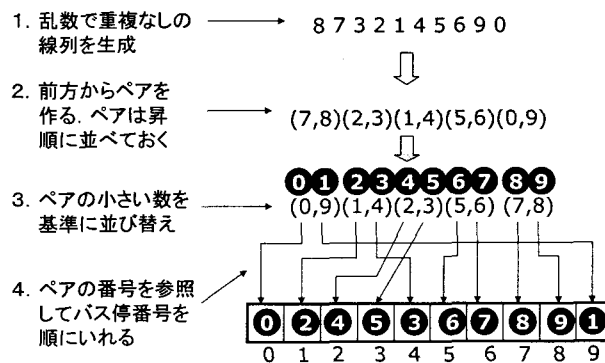


Fig. 8

Fig ??に示すように ID をつけることにより制約条件は、各 ID が重複しないこと、偶数 n より $n+1$ が後方に存在すること、偶数 n は線列上で照準に並んでいることとなる。次に Fig. 8 に示す手順に従い線列を生成する。

4.2. 評価関数の設計

解候補である各個体がもつ遺伝子情報を 4.1 のコーディングルールを用いて、デマンドバスの運行スケジュールに変更し、運用コスト、利用者満足度によって評価を行う。運用コスト MC は、

$$MC = \sum_i BC_i + \sum_i RC_i - \sum_j DP_j \tag{1}$$

と定義する。バス j の運行コストおよび走行費用(燃費から算出)をそれぞれ BC_i 、 RC_i とする。また、利用者を乗せることによる利益 DP_j を運用コストから減じる。さらに利用者の利便性を考慮するためデマンドの予約時刻 Dt_i との差 Lt_i を利用者コストとして加える。

$$CC = f(\sum_i Lt_i + Rt_i)$$

$$Lt_j = Dt_j - (Dt_i + Bt_{i,j}) \tag{2}$$

遅延時間 Lt_j を、一つ前のデマンド i の予約時刻 Dt_i にバスがデマンド i からデマンド j への移動時間 $Bt_{i,j}$ を足した値である予想到着時間からデマンド j の予約時刻 Dt_j との差とする。移動時間 $Bt_{i,j}$ を求めるための経路はダイクストラ法により一意に決定する。 Rt_i は、乗車遅延時間であり、最短経路で目的地へ向かった際の時間と実際のバスの経路による時間の差である。また、関数 $f(x)$ により遅延時間の総合計は、利用者コストに変換される。関数 $f(x)$ は、制約条件として機能し、非現実的な解の探索を回避するように GA を設定することができる。本研究では、ユーザーの遅延時間を最大遅延時間(デマンドを全て 1 台のバスで回った場合)で割った値に実際の運用コストをかけ、ユーザーコストとした。パラメータ設計は今後の課題としたい。

各個体は、運行コスト MC と利用者の利便性をコストに換算した CC を加えたものを評価値として持ち、これを最小化する。

4.3. システムの適用

本研究の有用性を確認するために実際の都市を例にデマンドバススケジューリングを行う。対象とする地域はデマンドバスの研究を行っている高知県中村市³である。本システムでは、バスを 1 台動かす際の諸経費を 2000 円、バスの平均時速を時速 15km、燃料費を 1 リットルあたり

90 円、燃費を 1 リットルあたり 3km とする。本研究で用いた対象地域を Fig.9 に、デマンドを Table 1 に示す。

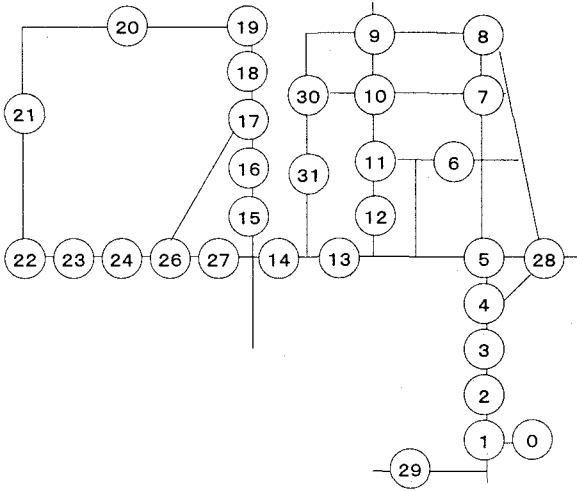


Fig. 9. Map of Nakamura city

Table 1. Demands

Demand ID	Due time	Bus stop
1	8:00	2
2	16:10	5
3	11:00	19
4	12:55	30
5	20:10	18
6	7:50	11
7	9:25	20
8	19:40	30
9	21:30	22
10	13:45	15
11	14:00	6
12	15:50	14
13	9:40	15
14	13:20	28
15	19:20	23
16	8:05	13
17	18:55	30
18	12:40	7
19	19:00	21
20	9:30	15

4.4. システムの実行結果

まずシステムの探索性能を検証するために、ユーザーの乗るバス停のみを考慮する設定で、実際の中村市のマップに Table 1 のデマンドを適用しシステムを実行した。Table 2 に GA 実行のパラメータを示す。Fig. 10 に実行結果を示す。図中の縦軸は 4.2 に従い求めた評価用コストで、横軸は世代数である。最良解の計算結果であるバススケジュールを Table 3 に示し、Table 4 に運用コストを示す。

Table 2 GA Parameters

Population	Cross over	Mutation	Generation
500	0.8	0.05	500

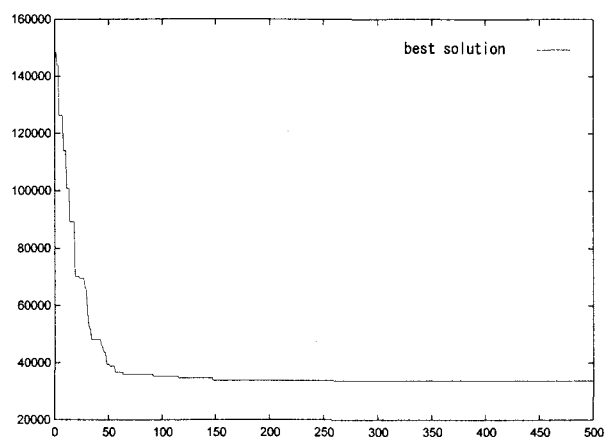


Fig. 10 Evaluation of GA

Table 3 Bus schedule

Bus 1	Time	7:50	8:00	8:05	
	Bus stop	11	2	13	
Bus 2	Time	9:25			
	Bus stop	20			
Bus 3	Time	9:30	9:40		
	Bus stop	15	15		
Bus 4	Time	11:00			
	Bus stop	19			
Bus 5	Time	12:40	12:55		
	Bus stop	7	30		
Bus 6	Time	13:20	13:45	14:00	
	Bus stop	28	15	6	
Bus 7	Time	15:50	16:10		
	Bus stop	14	5		
Bus 8	Time	18:55	19:00	19:20	19:40
	Bus stop	30	21	23	30
Bus 9	Time	20:10			
	Bus stop	18			
Bus 10	Time	21:30			
	Bus stop	22			

Table 4 Operating funds

Run cost	Bus cost	User Cost	Fee
2250	20000	19000	4000

Table 3 に示される実際の運行スケジュールを確認すると、おおむね実用的なスケジュールが作られていることが確認できる。実験ではデマンド数を 20 としたため利益ができるように運行するには、2 台以下のバスで運行する必要がある。しかしながらユーザーコストを導入することで、できるだけ利用者の利便性を向上しつつ安価な運用スケジュールを探索する結果になった。また Bus 2 と Bus 3 のように、距離が離れている場合、同時刻に別のバスによって運用するといった柔軟なスケジュールが作成できている事が確認できる。

次にユーザーの乗り降りを考慮した GA の性能を評価するためデマンド数を 2 倍の 40 に増やし実験を行った。10 回試行した最良性能の結果を Fig. 11 に示す。また

Table 5 に実行結果の一部を示す。各スケジュールとも制約条件を満たしていることが確認できた。

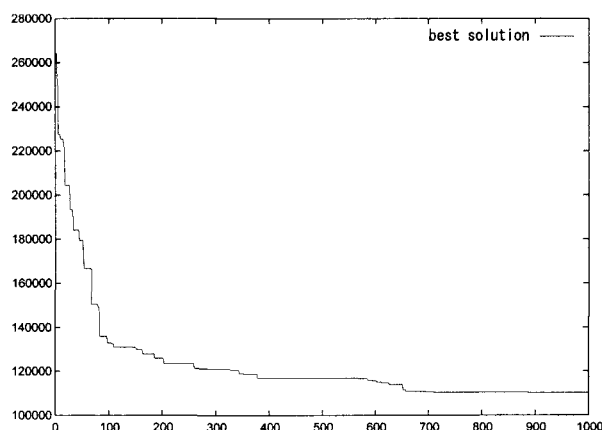


Fig. 11 Evaluation of GA

Table 5. Bus schedules

Bus 1	Time	8:15	18:16		
	Bus stop	31	30		
Bus 2	Time	19:00	19:41	19:23	19:34
	Bus stop	21	28	23	3
Bus 3	Time	16:10	16:16		
	Bus stop	5	0		
Bus 4	Time	8:05	8:15		
	Bus stop	17	28		
Bus 5	Time	14:00	14:11		
	Bus stop	6	17		
Bus 6	Time	16:00	16:12	16:25	16:34
	Bus stop	11	22	8	14
Bus 7	Time	13:45	13:50		
	Bus stop	15	13		
Bus 8	Time	8:00	8:02	8:10	8:24
	Bus stop	2	0	6	20
	Time	8:28	8:42		
Bus 9	Time	12:40	12:47		
	Bus stop	7	0		
Bus 10	Time	16:45	16:59		
	Bus stop	8	19		
Bus 11	Time	9:25	9:32	9:40	9:47
	Bus stop	20	14	15	28
Bus 12	Time	11:00	11:01	11:08	11:21
	Bus stop	15	26	28	29
Bus 13	Time	20:10	20:22	20:25	20:34
	Bus stop	18	5	8	14
Bus 14	Time	11:00	11:16		
	Bus stop	19	0		

5. 結 論

本研究では、改良型 GA を用いてデマンドバススケジューリングをバスの便数および各バスへの割り当てと巡回回路を同時に最適化するシステムの構築を試みた。更にデマンドバススケジューリングを表す線列に複雑な制約条件を有効に考慮できるコーディング方法を示した。有

用性を確認するために実際に中村市の地図を例にシステムの実行と検証を行った。まずユーザーの目的地を考慮せず、複雑な条件がないスケジューリングについてシステムの検証を行い、その結果、デマンド数が少ない小規模な問題の場合、有用な結果が得られた。更に、デマンド数を増やし問題規模を大きくし現実問題に近づけ、ユーザーの目的地を考慮に入れた実験で妥当な解を得ることができた。またデマンドバスに求められる要件を考慮して、ユーザーコストの概念を導入し、目的関数を決定することで利用者の利便性を確保しつつ、できるだけ運用コストを押さえるスケジュールの構築が可能となった。更に同時刻にバスを運行しデマンドを担当するといった柔軟なスケジュールも容易に作成できることが確認できた。

本研究では、問題を単純化させるためにバスの台数を限定せず、無制限に走らせるものと定義した。今後、実

用に耐えるシステムを開発するためには、一度に走れるバスの上限台数を考慮し、運行途中のデマンドにもある程度対応できるスケジューリングシステムを構築する必要がある。

参考文献

- ¹ 古田・杉本：遺伝的アルゴリズムの構造工学への応用，森北出版，1987
- ² 「GAにおける種々の工夫」 杉本博之・Lu Bianli
第11回設計工学・システム部門講演論文集2001, 11
- ³ 中村市バス
<http://www.city.nakamura.kochi.jp/news/bus/dbus.html>