

〔生物工学会誌 第71巻 第3号 171-178. 1993〕

遺伝的アルゴリズムによる発酵プロセスの最適軌道算出

松浦 一雄^{1*}・斯波 大幸¹・布川弥太郎¹・清水 浩²
塩谷 捨明²・菅 健一²

大関(株)総合研究所,¹ 大阪大学工学部応用生物工学科²

¹〒663 西宮市今津出在家町4-9

²〒565 吹田市山田丘2-1

(平成5年1月11日受付 平成5年3月12日受理)

Calculation of optimal trajectories for fermentation processes by genetic algorithm. KAZUO MATSUURA,^{1*} HIROYUKI SHIBA,¹ YATARO NUNOKAWA,¹ HIROSHI SHIMIZU,² SUTEAKI SHIOYA,² and KENICHI SUGA² (General Research Laboratory, Ozeki Corporation, 4-9 Imazu Dezaike-cho, Nishinomiya-shi, Hyogo 663¹; Department of Biotechnology, Faculty of Engineering, Osaka University, 2-1 Yamadaoka, Suita-shi, Osaka 565²) *Seibutsu-kogaku* 71: 171-178, 1993.

A genetic algorithm (GA) was improved in terms of mutation and was applied to obtaining optimal trajectories which gave the solutions to three problems: (1) finding the optimal trajectory of the specific growth rate for the maximum production of glutathion (GSH) in yeast fed-batch culture at a given operation time; (2) the same as (1), but with the additional constraint that the concentration of GSH should be higher than a certain value; (3) finding the optimal temperature trajectory for isoamylacetate formation in batch sake brewing. Problems (1) and (2) have been already solved by the Maximum Principle (MP), and problem (3) by Dynamic Programming (DP). GA gave sufficiently practical solutions that correlated with the solutions obtained by MP and DP, and GA was shown to be useful for finding the optimal trajectory with a shorter calculation time.

発酵生産物を経済的に作るには、できるだけ合理的で最適な設計、運転をする必要がある。これら発酵プロセスの最適化問題においてグリーンの定理、最大原理、ダイナミックプログラミング、シンプレックス法など、いくつかの方法が用いられてきた。また、パラメータ変動に対処するため、発酵期間中に同定変更したパラメータを用いて、その度ごとに最適化計算を行う Kishimoto らの示した方法¹⁾が報告されている。このようにオンラインで最適化計算を行わせようとした場合、最適化計算が高速であることが望まれる。Kishimoto らはダイナミックプログラミングを用いて、関数再帰方程式を最終時点から順に解いていき、計算範囲内でのすべての状態点において、その時点から最終培養時点までに生産されるグルタミン酸量の最

大値を培養開始前にあらかじめ計算しておき、培養期間中において実測データが示す状態点にその時点の計算された最適軌道上の状態点を入れ換えた後、次の計算時点までの最適政策ならびに最適な培養経過を計算させるようにし、実測値に対応させながら最適化計算の高速化を計っている。

ところが、この方法ではあらかじめ計算しなければならない状態点の範囲は膨大な量であるし、繁雑である。状態点を記憶させておく大きなメモリも必要である。望ましくは、一つのプログラムで、プロセスの構造を記述したモデルがあり、また培養期間中に得られた実測値から高速に最適化計算が可能な方法があれば良い。加えて最近プロセスの状態を記述するために、ファジィ理論²⁾やニューラルネットワーク³⁾を使用した例が増えているが、これら不連続で複雑な関数においても最適化計算が可能であることが望まれる。

* 連絡先, Corresponding author.

そこで我々は、近年、組合せ最適化問題において威力を発揮することが示されている遺伝的アルゴリズムに着目して、前記の条件を満たす最適化アルゴリズムとなりうるかどうかを検討した。すなわち、遺伝的アルゴリズムによる発酵プロセスの最適軌道算出への適用可能性、および適用に際しての問題点とその発酵プロセスに適した改良を試みた。

遺伝的アルゴリズムは組合せ最適化問題の解を求めるのに主として用いられ、プラントオペレーションにおいて、操作時間、コスト等を最小にするジョブジョブスケジューリング問題⁴⁾やフロージョブスケジューリング問題⁵⁾、アミノ酸配列の中から目的の配列を検索する問題⁶⁾、バックプロパゲーションの重み係数を決定すること⁷⁾などに応用されている。今回われわれが対象とした汎関数の最適化も、組合せ最適化問題の一種として考えればこの方法が適用できると考えられた。

実験方法

最適化問題と二点境界値問題 汎関数の最適化問題は、状態ベクトルを X 、操作変数のベクトルを u としたとき、プロセスの状態が次の常微分方程式、

$$\begin{aligned} dX_i/dt &= f_i(X, u) \quad (i=1, 2, \dots, N) \\ X_i(t_0) &= a_i \quad (a_i: \text{const.}) \end{aligned} \quad (1)$$

で表されたとき、評価関数の一般形として、

$$J = \int_{t_0}^{t_f} f_0(X, u) dt \quad (2)$$

を最大または最小にするいわゆる変分問題として定式化できる。⁸⁾以下に本報告で取り組んだ3つの問題について示した。

1) 半回分培養におけるグルタチオン生産量を最大にする問題 グルコースを単一炭素源とする半回分培養において、清水ら^{9,10)}は次のような生成モデルを報告している。菌体増殖は、

$$d(Vx)/dt = \mu Vx \quad (3)$$

で示され、 V は培養体積 (l)、 x は菌体濃度 (g/l)、 μ は比増殖速度 (h^{-1})である。また、菌体内グルタチオン含量は、

$$dp/dt = \rho_G - \mu p \quad (4)$$

で示され、 p は菌体内グルタチオン含量 ($mg\text{-GSH}/g\text{-cell}$)であり、 ρ_G はグルタチオン比生産速度 ($mg\text{-GSH}/g\text{-cell}\cdot h$)

であり、生産と増殖による希釈の差で表される。このとき、グルタチオン比生産速度 ρ_G は μ によって次の式により示される。

$$\begin{aligned} \rho_G &= 11.7836 \mu \quad (0 \leq \mu \leq 0.2932 \text{ h}^{-1}) \\ \rho_G &= 6.3585 - 9.9025 \mu \quad (0.2932 \leq \mu \text{ h}^{-1}) \end{aligned} \quad (5)$$

この関係は、グルタチオン比生産速度が最大となる比増殖速度 μ_c を頂点とした、二つの比増殖速度の一次式で表されている。

また、グルタチオン総量 pVx の変化は、

$$d(pVx)/dt = \rho_G Vx \quad (6)$$

と与えられる。したがって、以上のように定式化されたモデルを用いて、発酵終了時点で評価関数、

$$J_1 = pVx(t_f) \quad (7)$$

を最大にする比増殖速度軌道を求めるのがここで考える問題である。

2) 二点境界値問題^{9,10)} この問題は発酵終了時点で菌体内グルタチオン濃度を設定値にしつつ、これを束縛条件として(7)式で示される評価関数を最大にする比増殖速度軌道を求めるものである。

3) 清酒中の酢酸イソアミル濃度を最大にする問題¹¹⁾ 糖化液を用いた清酒の仕込において、発酵終了時点で酢酸イソアミルを最大にする温度軌道を求める問題である。菌体増殖はLeeらのモデル¹²⁾

$$dx/dt = \mu_m(1-x/x_m)x \quad (8)$$

とした。ここで、 μ_m 、 x_m は μ と x のそれぞれ最大値であり定数である。アルコール生成は南波らの積算エタノールモデル¹³⁾にて、

$$dP/dt = \alpha dx/dt + \beta_0 \exp(-k_d \int_0^t P d\tau)x \quad (9)$$

ここで、 P はエタノール濃度 (g/l)、 α 、 β_0 、 k_d は定数である。

酢酸イソアミル濃度は、

$$dIAA/dt = Y_{IAA/P} dP/dt - k_d' IAA \quad (10)$$

として示される。ここで、 IAA は酢酸イソアミル濃度 (g/l)、 $Y_{IAA/P}$ はエタノールに対する酢酸イソアミル収率 ($g\text{-IAA}/g\text{-ethanol}$)であり、 k_d' は分解係数 (h^{-1})である。パラメータ μ_m 、 x_m 、 β_0 、 k_d 、 $Y_{IAA/P}$ 、 k_d' は前報に示したように温度に影響を受けたので、温度を説明変数とする多項式で表した。¹¹⁾(8)-(10)式は、発酵初期には

温度が高いほど酢酸イソアミル生成速度は大きくなるが、発酵後期に酢酸イソアミル分解速度も大であり、温度が低い方が酢酸イソアミルが保存されることを示している。評価関数は発酵終了時点の酢酸イソアミル濃度である。

$$J_2 = IAA(t_f) \quad (11)$$

使用計算機 今回作成したプログラムは Turbo-C (Ver. 2.0, ボーランド社) で構築し、エプソン 486GR を用いて計算した。

遺伝的アルゴリズム 遺伝的アルゴリズム¹⁴⁾ (遺伝アルゴリズムとも呼ばれる) は生物学者が生物の進化適応をシミュレーションするために発展させてきたものを Holland ら¹⁷⁾ が体系づけたもので、生物の世代交代、染色体の交叉、突然変異、個体の選択という過程を模した操作を行うことにより、より環境に適応した個体を発見していく方法である。個体の持つ遺伝情報を我々の問題における軌道の組み合わせととれば、組み合わせの進化、すなわち最適解の探索が行える。またここで言う環境とは、我々が問題に課した束縛条件や評価関数のなんらかの尺度である。Fig. 1 にそのアルゴリズムの概略を示した。これは Simple GA と呼ばれるものであり、コーディング、適応度の計算、選択複製、交叉、突然変異の各ステップからなる。Fig. 1 に沿って操作手続きを以下に示した。

1) コーディング Fig. 2 に示すように発酵開始 t_0 から t_f までを M 分割し、おのおのの段における決定されるべき操作量 u_i ($i=0, 1, \dots, M-1$) のとりうる離散的な数値を格納した配列のカウンタ用変数を遺伝

子としてコーディングし、これを各段ごとに繋ぎ合わせて染色体とした。このとき、グルタチオンの問題では政策 u_i として μ の時系列を決めることとし、 μ_c を含む離散的な μ_{\min} から μ_{\max} までの値を20個、酢酸イソアミルの問題においては操作温度の時系列を決めることとし、10°C から 30°C までの 1°C おきの21個の値をおのおの用意した。

厳密に最適値を求めるのであれば操作変数を2進数等でコーディングし、任意の実数を扱うべきであるが、ここでは実用を考慮して、早く解の近傍に到達することを優先させて操作変数は離散値にのみ限定し近似解を得ることを目的とした。結果を見ても判るように、今回の問題では離散値のみを使用しても充分正確な解答が得られた。むしろ微生物反応の不確定性を考慮した場合、厳密に求めすぎても得られる利得は労力に比してごくわずかであると考えられる。

2) 初期値 初期値は乱数を用いて決定した。このとき、異なる染色体 (ここでは政策 u_i の配列) を20個用意し、後の操作を行った。遺伝的アルゴリズムの利点の一つは、この染色体の数だけ解空間を多点探索できることである。¹⁵⁾ 以降、個体 (individual) と染色体 (chromosome) はほぼ同様の意味であるが、個体と染色体の関係は、染色体がコードしている遺伝子を交差させたり突然変異させたりし、染色体が表現型に翻訳されて政策 u_i の配列を持つ個体となる。

3) 適応度の計算 グルタチオン、酢酸イソアミ

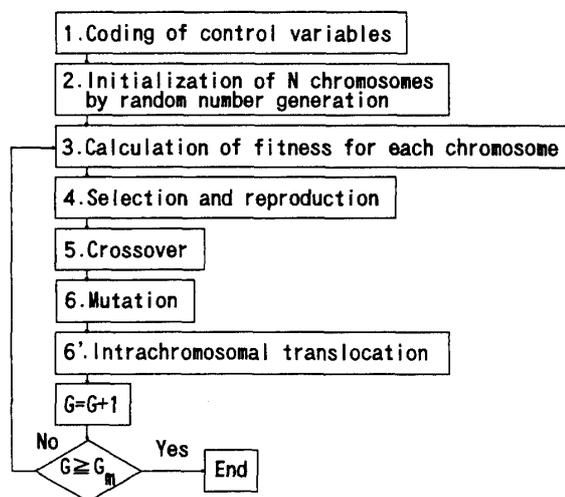


Fig. 1. Genetic algorithm calculation procedure.

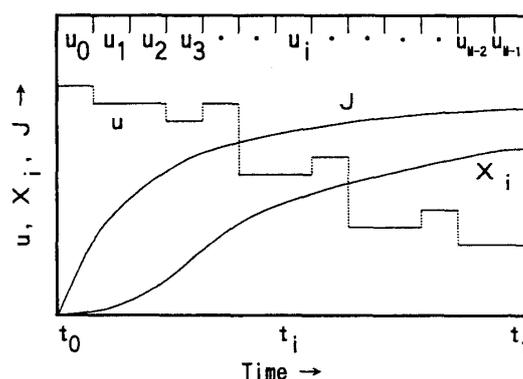


Fig. 2. Method of coding control variables. The fermentation time from t_0 to t_f was divided into M stages. Discrete data (u_i) of control variables were initialized by the generation of random numbers and chromosomes as genes were made from the data set of $\{u_i\}$. The chromosomes were altered by the genetic algorithm, that is, by decoding the control variables and substitution in the model, and the fitness of individual chromosomes was evaluated.

ルの濃度を最大にする問題では、評価関数 (7), (11) 式をそのまま用いた。二点境界値問題では、我々は次のように適応度 f_1, f_2 を定義した。

$$f_{1,G} = J_{1,G} - (J_{1,G-1,\max} / \varepsilon_{G-1,\min}) (G/G_{\max}) \varepsilon_G \quad (12)$$

$$f_{2,G} = J_{1,G} - C \varepsilon_G \quad (13)$$

ただし、

$$\varepsilon_G = |p(t)_G - p_f| \quad (14)$$

$$J_{1,G} = p V x(t)_G \quad (15)$$

ここで、 ε は発酵終了時点のグルタチオン含量 $p(t)$ のその設定値 p_f に対する誤差であり、 J_1 は評価関数である。 f_1 は第一の適応度であり評価関数から誤差に世代数 (繰り返し計算の回数) G とその最大値 G_m 、および前の世代の J_1 の最大値 $J_{1,G-1,\max}$ と ε の最小値 $\varepsilon_{G-1,\min}$ の比率を乗じてペナルティを課している。また下添字 G は世代数 G での値を意味する。(13) 式のみでは係数 C の値が小さすぎると誤差量 ε を無視する傾向にあり、 C の値が大きすぎると J_1 のより高い値が得られにくい傾向にあった。そこで、最初世代数が小さいときは (12) 式第二項はほとんど無視され、評価関数 J_1 のみに支配されることにより J_1 に対する最適解を求めようとするが、世代を経るにつれて緩やかに束縛条件 ε を満たすように J_1 と ε の値が等しくなるように前の世代の結果をフィードバックさせている。このスケジューリングによって ε 最小化の拘束のもとに J_1 の最適化を安全に行うことを意図した。 f_2 は第二の適応度であり、パラメータ C は固定して与えた。 f_2 を全世代を通じた絶対的な評価基準とした。

4) 選択複製 まず個体群のなかでもっとも高い適応度の95%以下の適応度を示す個体の適応度は強制的に適応度を0とし、複製させないこととした。すなわち、

$$f_j' = f_j - 0.95 \cdot f_{\max}$$

ただし

$$f_j' < 0 \text{ ならば } f_j' = 0 \quad (16)$$

さらに、通常ルーレットルールと呼ばれる次式に基づいて複製個数を決定した。ある個体の複製個数 n は、

$$n = N f_j' / \sum_j f_j' \quad (17)$$

により得られる。 N は全個体数であり、 f_j' は j 番目の

個体の (16) 式による淘汰後の適応度である。(17) 式は適応度の総和における各個体の適応度の占める割合に応じて n 個だけ次の世代へその複製を残すことを示している。 n は当然整数値でなければならないので、(17) 式の右辺の小数点以下を四捨五入した。二点境界値問題においては f_1 を使用して (16) (17) 式から複製個数を求め、 f_2 に基づいて絶対的にすぐれた個体を必ず個体群の中に入れることとした。過去に一度適応度を計算した個体ではあるが、この後行交差、突然変異の操作により別の遺伝子を持つようになる可能性が高く、良い遺伝子を系の中に保存する効果も期待できる。

5) 交差 交差を行わせる前に、複製を行った後の各染色体について親の異なるもの2つずつのペアを作り、乱数を発生させて交差を開始する位置 (遺伝子座) の遺伝子を決定し、それ以降の同じ位置の遺伝子を互いに交換する操作が交差である。たとえば、操作変数の配列のカウント用変数が遺伝子となっているので、次の二つの染色体について、

chromosome 1: 2 5 7 3 | 1 2 5 4 6

chromosome 2: 8 7 9 5 | 3 7 5 2 1

点線の箇所では交差が起こった場合、二つの染色体は

chromosome 1: 2 5 7 3 | 3 7 5 2 1

chromosome 2: 8 7 9 5 | 1 2 5 4 6

と書き換えられる。この操作の意味は、染色体群の組み合わせの多様性を増すことにある。¹⁴⁾ 交差はある確率 P_{cross} のもとに行う。

6) 突然変異 ある確率 P_{mutation} のもとに染色体を選択し、その染色体上のある遺伝子座を同様に乱数により決定する。そして、その遺伝子を別のコードを持った遺伝子に置き換える。たとえば、上記染色体1の下線部に突然変異が起こった場合、

chromosome 1: 2 5 7 3 3 7 5 2 1

乱数によって選ばれた新しい値に置き換えられる。

chromosome 1: 2 5 7 5 3 7 5 2 1

突然変異は常に解空間をランダムに探索することにより、局所解に陥ることを防いでいる。¹⁴⁾ これを後述する変異と区別するためランダム突然変異と名づけておく。

6') 転座 通常転座はあまり用いられないよう

であるが、今回グルタチオンの問題^{9,10)}に用いた。ここでは、連続した3つの遺伝子を乱数により選びだし、これを別を選びだした同じ染色体上の連続する3つの遺伝子とそっくり入れ換えてしまう操作である。

chromosome 1: 3 6 4 6 2 7 2 1 9

下線部分を入れ換えて、

chromosome 1: 3 2 1 9 2 7 6 4 6

この操作も染色体群の多様性を増す。¹⁶⁾

ここまでの操作を行った後、再び適応度を計算し、計算を繰り返す。この一連の計算過程を一代と呼ぶ。

中間値突然変異 通常発酵プロセスの制御においては、急激にその環境を変化させる回数はあまり多くない。境界値制御やたとえば温度感受性プロモーターのスイッチをオンオフする操作があるが、従来の経験上は多くて二度程度であり、それ以外では一定値をとらせたり、連続した操作を行わせることが多い。そこで、操作が連続して生ずることを意図して、ある遺伝子を乱数によって選びだし、その両隣の遺伝子が持つ比増殖速度あるいは温度の中間の値になるような変異をあわせて行った。これを中間値突然変異と名づけた。

たとえば、

chromosome 1: 3 6 4 6 2 7 2 1 9

下線部の遺伝子が乱数により選択されたとき、

chromosome 1: 3 6 4 3 2 7 2 1 9

のように操作し、両隣の平均値をとり小数点以下を切り上げて新しい遺伝子とした。

結果と考察

中間値突然変異の効果 グルタチオン総生産量を最大にする問題およびその二点境界値問題は、最大原理によりすでにその最適軌道が算出されている。最大原理による前者の軌道は、Fig. 4 に点線で示した Bang-Bang 型になることが解っている^{9,10)}が、これを遺伝的アルゴリズムにより解いた。Fig. 3 は最大原理による評価関数値を100として、その相対値の計算経過を示している。Fig. 3 中の数字は遺伝子(段)数を10としたときの、確率 P_{mutation} のもとに行う中間値突然変異の数であり、中間値突然変異とランダムな突然変異が必ず合計して10個になるように変異させている。このとき、初発菌体濃度は1g/l、初発グルタチオ

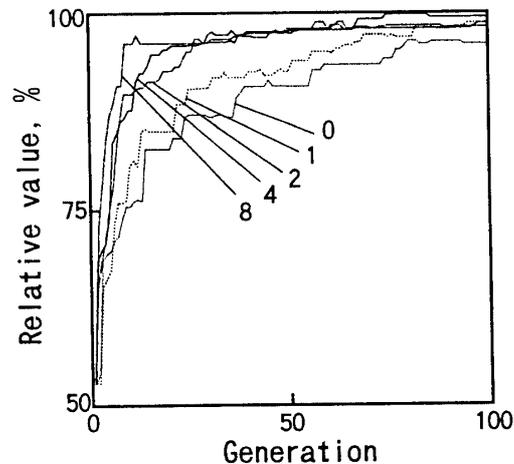


Fig. 3. Effect of intrachromosomal mutation by intermediate value on the optimization procedure. The problem of the maximization of total glutathion production was solved at various numbers of intrachromosomal mutation by the intermediate values indicated in the figure. The value of the objective function by MP was taken as 100%.

ン含量は5.24 g-GSH/g-cell、終了時間は10時間、 $\mu_{\text{max}}=0.42$, $\mu_c=0.2932$, $\mu_{\text{min}}=0.05$ (h^{-1})とし、文献⁹⁾に従った。式(3)-(6)の解はRunge-Kutta-Gill法を用いて解き、ステップ幅は0.5hとした。また最初遺伝的アルゴリズムにおいて転座は行わなかった。交差確率は $P_{\text{cross}}=0.5$ 、変異確率 $P_{\text{mutation}}=0.5$ としたが、これらの確率は予備的な試行計算により決定した。一般的には交叉確率は1に近く、変異確率は0.1以下にしている例が多い。突然変異はボルツマンマシンにおけるシミ

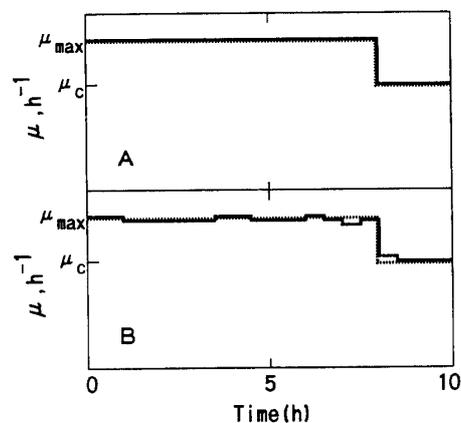


Fig. 4. Effect of intrachromosomal translocation on the optimization procedure. Dotted lines indicate results by MP. Solid line A shows the result by the genetic algorithm with translocation, and solid line B without translocation.

エレーテッドアニーリングのように、局所解に落ち込むことを防ぐためにある程度誤差が増大することを許し、許容誤差を少しでも小さくすることで、最適解を求めようとする考えに近い¹⁶⁾と言われるが、突然変異の確率が高すぎるとかえって良い遺伝子が失われる可能性が高くなるとされている。Fig. 3の結果は中間値突然変異の数が増えるのに従って、より早く解が求まる傾向にあった。ただし、その数が8になると、100%に近づいた付近でかえって良い情報が失われる傾向にあった。これは境界値から境界値へのシャープな切り替えが発見されにくく、もしくは発見されても失われ易くなっていると思われた。いずれにしても、中間値突然変異をある程度多く起こさせた方がよいということは、この変異により良い遺伝子の小さなまとまり、Hollandの言うビルディングブロック¹⁷⁾が得られ易くなっているためと考えられた。

転座の効果と欠点 転座の有効性を、まずグルタチオン総生産量を最大にする問題⁹⁾を対象として論じた。Fig. 4に転座を行わせた結果(A)と、行わせない結果(B)を示した。いずれも中間値突然変異の数は4であり、転座を行わせる確率は0.25とした。その他のパラメータはFig. 3と同様とした。最大原理においては、 μ_{max} から μ_c への切り替え時間は7.95時間後であるが、転座を行わせた場合8時間後に切り替えており、ほとんど完全な解が得られた。一方転座を行わなかった場合には、ほぼ正しい解が得られているが、比増殖速度の軌道が上下しており、一つの染色体上で遺伝子の対を交換する転座の有効性が確認された。しかし、これはある期間一定値を取らせるという解の構造によるものであることが充分予想される。

そこで、Fig. 5に糖化液を用いた清酒の醸造において酢酸イソアミル濃度を最大にする温度軌道を求める問題の計算結果を示した。Aは転座と中間値突然変異を行わせたもの。Bは中間値突然変異のみ、Cは転座のみを行わせている。このときの計算条件は中間値突然変異数4、200世代学習させたもので、その他のパラメータはFig. 3と同様である。菌体数の初期値は0.446 g/l (2×10^7 cells/ml)、エタノール、酢酸イソアミル濃度の初期値は0とし、発酵終了時間は138時間とした。Fig. 5中点線は同様の問題をダイナミックプログラミングにより解いた軌道である。¹¹⁾ Fig. 5に示した結果のごとく、中間値突然変異を起こさせたもの(A, B)は、ほぼ正しい結果が得られているのに対し、転座のみ(C)では温度の上下が大きく、充分とは言え

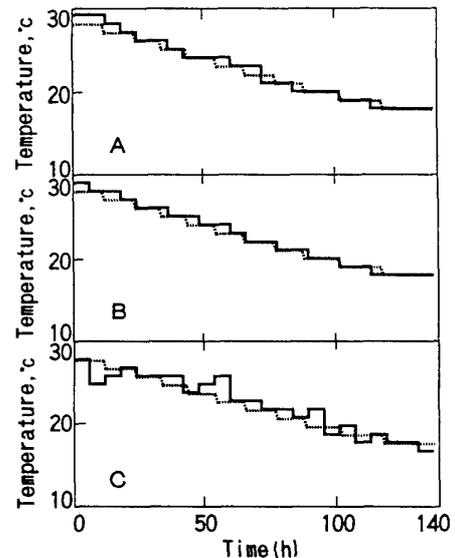


Fig. 5. Effect of intrachromosomal translocation on the maximization of isoamylacetate. Dotted lines indicate results by dynamic programming (DP). Solid lines show the results by the genetic algorithm: A, with translocation and intrachromosomal mutation by intermediate value; B with intrachromosomal mutation by intermediate value and without translocation; and C without both mutation by intermediate value and translocation.

なかった。このように、酢酸イソアミルの最適化問題においては、転座の効果はあまりなく、解がほぼ得られているのは中間値突然変異に負うところが大きい。ある点で対称な軌道を持たない解では、むしろ逆効果になることが容易に理解される。したがって、転座は解の構造を充分吟味した上でなければ使用にあたり注意を要すると思われた。これに対して、中間値突然変異はある程度連続した解であれば、発酵プロセスの最適化に非常に有効であると考えられた。

二点境界値問題の解 発酵終了時点で菌体内グルタチオン含量を設定値にし、その束縛条件のもとにグルタチオン総生産量を最大にする二点境界値問題の計算結果を示した。まず、束縛条件を考慮した適応度を式(13)の f_2 のみで評価する場合を考えた。ここで、パラメータCは固定した値をとり、この値の影響をみた。結果をFig. 6に示した。世代数は500であり、中間値突然変異、転座などのパラメータはFig. 4 Aの場合と同様とした。Cの値が小さいときには、誤差量 ϵ をまったく考慮しておらず、最大化問題と同様の結果となっている。Cの値を大きくするにつれてその拘束条件が厳しくなり、正確な結果が得られた。しかし、Cの値が大きい場合には誤差量を小さくしようと

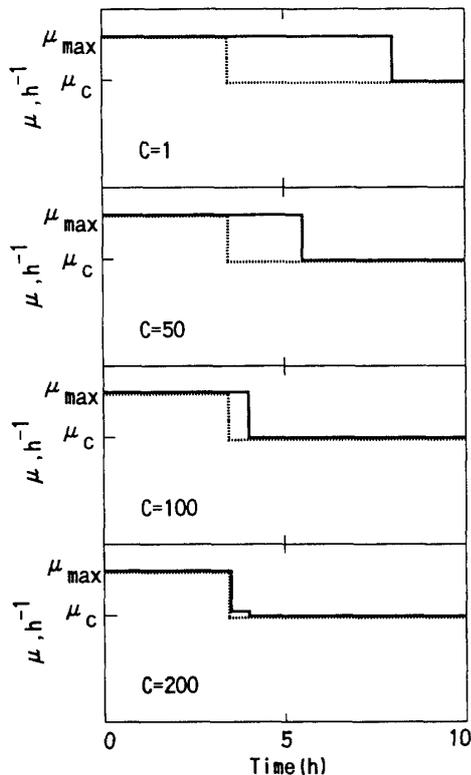


Fig. 6. Calculation results of two-point boundary value problem of GSH production. Dotted lines indicate results by MP. Solid lines indicate the results by GA, in which fitness was defined as $f = J_1 - C\epsilon$, where C was fixed at the values indicated in the figure. Mutation by intermediate value and translocation were employed in this calculation.

して、グルタチオン総生産量 J_1 の値を無視する傾向にあり、転座や中間値突然変異を行わなかったときには、正しい結果が容易に得られなかった。

次に、式(12)-(15)を用い、第一の適応度 f_1 により複製個数を(16)式により決定し、第二の適応度 f_2 により全世代を通してもっとも高い適応度を示す個体を必ず残すようにした場合に得られた結果を Fig. 7 に示した。このとき C の値を100とした。この結果 Fig. 6 に比較して、より正しい結果に近づいていることがわかる。このような段階的な適応度による評価が好ましいことがわかった。

この二点境界値問題は酒類の発酵工程においても非常に重要な意味を持っている。酒類の発酵はエタノールの生成量あるいは濃度を重要な指標とするが、二点境界値問題を解くことにより、エタノール濃度(生成量)を発酵終了時点で目標とする設定値にして、かつ他の成分、官能に影響を及ぼす酢酸イソアミルやその他の香気成分、有機酸の濃度などを最適に生成させる

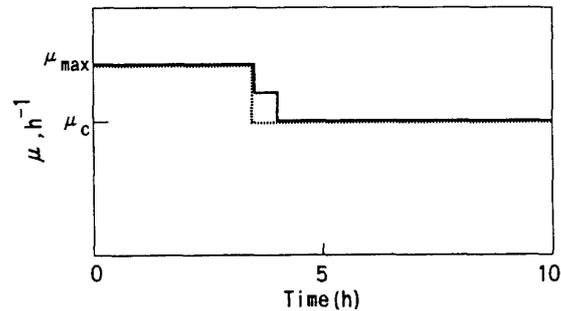


Fig. 7. Calculation results of two-point boundary value problem of GSH production. Solid and dotted lines have the same indications as those in figure 6. Fitness was defined by eqs. (12)-(15).

ような軌道を計算することが考えられる。

計算時間 今回使用した言語はCであり、インテル社の486CPUに487の数値演算プロセッサを装着した環境で計算したが、グルタチオンの最適化問題で約30秒、二点境界値問題で約3分、酢酸イソアミルの最適化問題で1分強であった。処理系とハードウェアの違いもあるが、実用範囲内であり、オンラインにおける最適化計算を行わせるに充分短時間であった。

遺伝的アルゴリズムの問題点 一つは、コーディングの問題であり、利用者の主観やセンスに負うところが大きく、やり方はさまざまである。今回の問題でも操作変数の配列のカウン用変数を符号なし整数値としてそのままコーディングしたが、他に2進数やグレーコードも考えられる。第二に、パラメーターの数が多いことである。交差、変異の確率や計算の打ち切り世代数、淘汰するとき足切りを行うならその値もそうであり、問題ごとに検討されなければならない。¹⁵⁾

今回対象とした問題はいずれもすでに結果が求められ、実験的に確認されているものであったが、パラメーター決定は厳密な予備的計算を行ったわけではなく、いずれの問題でも意図的に共通した値とした。未知のプロセスにおいて、オンラインで計算する前に段数を少なくして試行的に計算して解の構造を把握しておくことは可能であり、遺伝的アルゴリズムは、かなりの普遍性を持つものであると思われた。また、今回対象としたいずれの問題も実験的にその最適性が実証されていることをつけ加えておく。

要 約

1. 遺伝的アルゴリズムにより、発酵プロセスの最適軌道算出を試みた。半回分培養におけるグルタチオ

ン総生産量を最大にする問題と、その発酵終了時点で菌体内グルタチオン含量を設定値にする束縛条件のもとに総生産量を最大化する二点境界値問題、糖化液を用いた清酒の発酵プロセスにおいて発酵終了時点の酢酸イソアミル濃度を最大にする問題を解き、遺伝的アルゴリズムの実用性を検討した。

2. グルタチオンの問題の最大原理による結果と酢酸イソアミルの問題のダイナミックプログラミングによる結果に対して、遺伝的アルゴリズムではいずれもほぼ同様の軌道が実用的時間内に算出できた。

3. 遺伝的アルゴリズムにおいて、ある遺伝子の持つ制御変数を、両隣の遺伝子の持つ制御変数の中間値にさせる中間値突然変異を導入したところ、計算時間が短縮され、我々が提案した中間値突然変異が発酵プロセスの最適化において有効であることが示唆された。グルタチオンの問題では転座という一つの染色体上で遺伝子対を入れ換える操作を行ったことにより、より正しい結果が得られた。しかし、転座は酢酸イソアミルの問題ではかえって逆効果であった。

記 号

C : coefficients, —
 f : function represented relation between state vector and control variable vector
 f_1, f_2, f_j : fitness, —
 G : generation, —
 IAA : isoamylacetate concentration, g-IAA/l-sake
 J : objective function
 J_1 : objective function on optimization procedure of glutathion, g-GSH
 J_2 : objective function on optimization procedure of isoamylacetate, g-isoamylacetate/l
 M : number of stage, —
 n : number of reproduced individuals, —
 N : number of total individual, —
 p : glutathion concentration in cell, g-GSH/g-cell
 P : ethanol concentration in sake, g/l
 t : time, h
 u : control variable
 V : volume of broth on fed-batch culture, l
 x : cell concentration, g/l
 X : state vector
 $\alpha, \beta_0, k_d, k_d', Y_{IAA/P}$: constants

ϵ : error between the calculated glutathion concentration in yeast and the set value, g-GSH/g-cell
 μ : specific growth rate, h⁻¹
 μ_c : optimum specific growth rate for maximum specific glutathion production rate, h⁻¹
 ρ_G : specific glutathion production rate, g-GSH/g-cell·h⁻¹
 <subscripts>
 i, j : counter variable
 0 : initial
 f : final
 \max, m : maximum
 \min : minimum

文 献

- 1) Kishimoto, M., Yoshida, T., and Taguchi, H.: *J. Ferment. Technol.*, **59**, 125-129 (1981).
- 2) Oishi, K., Tominaga, M., Kawato, A., Abe, Y., Imayasu, S., and Nanba, A.: *J. Ferment. Bioeng.*, **72**, 115-121 (1991).
- 3) 松浦一雄, 広常正人, 浜地正昭: *醸酵工学*, **69**, 463-469 (1991).
- 4) 西川 一, 玉置 久: *計測自動制御学会論文集*, **27**, 593-599 (1991).
- 5) 三宮信夫, 飯間 等: *日本機械学会第二回 FAN シンポジウム講演論文集*, p. 389-394 (1992).
- 6) 小長谷明彦: *日本機械学会第二回 FAN シンポジウム講演論文集*, p. 375-380 (1992).
- 7) ウィリアムス・ヴィクトル, 徳成 剛, 松岡清利: *電子情報通信学会 技術研究報告*, **91**(411), 93-99 (1992).
- 8) 吉田敏臣: *バイオエンジニアリング*, 日本醸酵工学会編, p. 51-73 (1985).
- 9) Shimizu, H., Araki, K., Shioya, S., and Suga, K.: *Biotechnol. Bioeng.*, **38**, 196-205 (1991).
- 10) 塩谷捨明: *醸酵工学*, **70**, 395-404 (1992).
- 11) Matsuura, K., Hirotsune, M., Hamachi, M., and Nunokawa, Y.: *J. Ferment. Bioeng.*, **74**, 112-116 (1992).
- 12) Lee, J. M., Pollard, J. F., and Coulman, G. A.: *Biotechnol. Bioeng.*, **25**, 497-511 (1983).
- 13) Nanba, A., Nishizawa, Y., Tsuchiya, Y., and Nagai, S.: *J. Ferment. Technol.*, **65**, 277-283 (1987).
- 14) Goldberg, D. E.: *Genetic Algorithm in Search.*, Addison-Wesley (1989).
- 15) 竹内 勝: *コンピューターソフトウェア*, **8**, 416-424 (1991).
- 16) 和田健之介: *Computer Today*, **47**, 49-61 (1992).
- 17) Holland, J. H.: *日経サイエンス*, Vol. 9, 56-65 (1992).