

付録D シミュレータの操作方法

D. 1 概要

本シミュレータは、エージェントがゲームする場を実装しただけでなく、ゲーム後に得られる類推マトリックスを解析する機構も組み込んでいる。この付録では、作成したシミュレータの開発環境、動作環境について述べ、ついでテストケースを踏まえて操作方法を説明する。

D. 1.1 開発環境

本シミュレータは Microsoft の統合開発環境 Visual Studio Version 6.0 を通じ、同社の Visual C++ Version 6.0 にて開発した。

D. 1.2 動作環境

本シミュレータの動作環境は表D. 1に示すとおりである。

CPU	Pentium II 400M Hz
メモリ	128 M byte
OS	Windows 98
ディスプレイ	1024×768 フルカラー

表 D. 1 動作環境

D. 2 シミュレーション

以降では簡単なテストケースを取り上げ、シミュレータ(何故か TW と呼ぶ)を使った実験手順を説明していく。あらかじめその進行を大まかに述べると、以下の表D. 2のようになる。

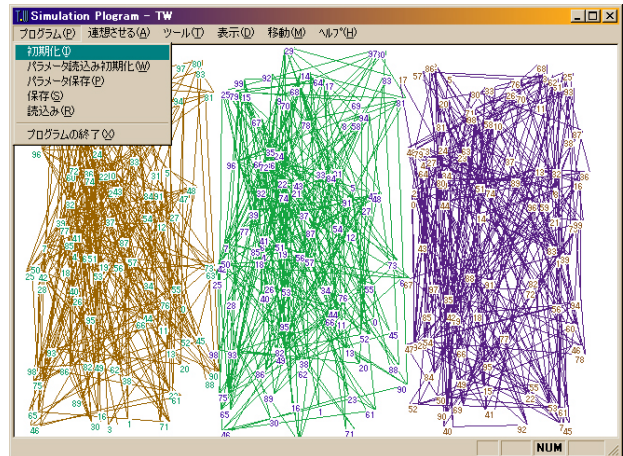
1	初期化
2	ゲーム(ターン回数指定)
3	データ解析(デンドログラム表示)

表 D. 2 実験手順

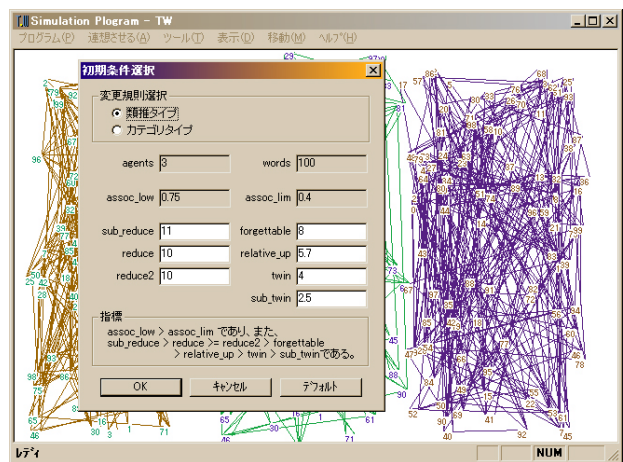
D. 2.1 初期化

初期化ではパラメータの調整を行う。

図D. 1では初期化する為に選択すべきメニューが示されている。メニューで「初期化」を選択すると、図D. 2のようなパラメータ設定ダイアログが現れる。必要とあれば、このときにパラメータの値を変更する。



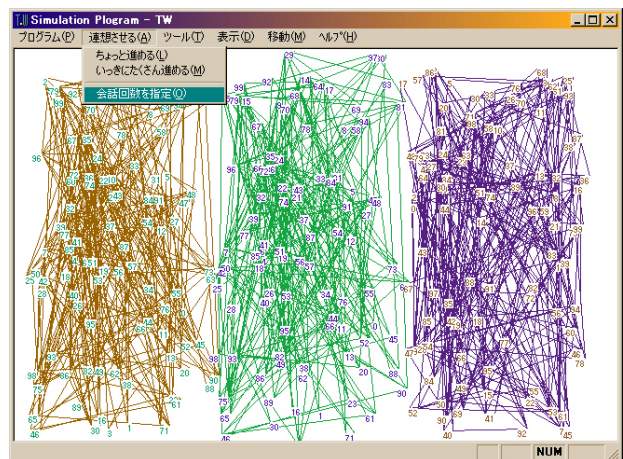
図D. 1 初期化するときを選択するメニュー



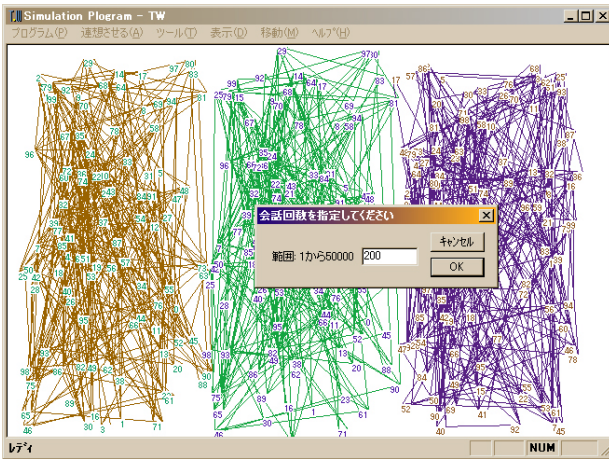
図D. 2 パラメータの設定ダイアログ

D. 2.2 ターン回数の指定

パラメータの設定を終えて「OK」を押すと、ダイアログが消えて初期化が完了する。ついでターン回数を指定し、エージェントにゲームを行ってもらうわけだが、そのターンの回数を指定するには下図D. 3のように、メニューを選択してから行う。

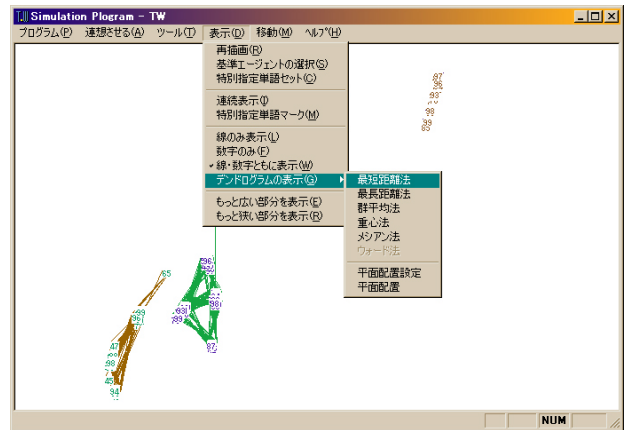


図D. 3 ターン回数指定メニュー位置



図D. 4 パラメータの設定ダイアログ

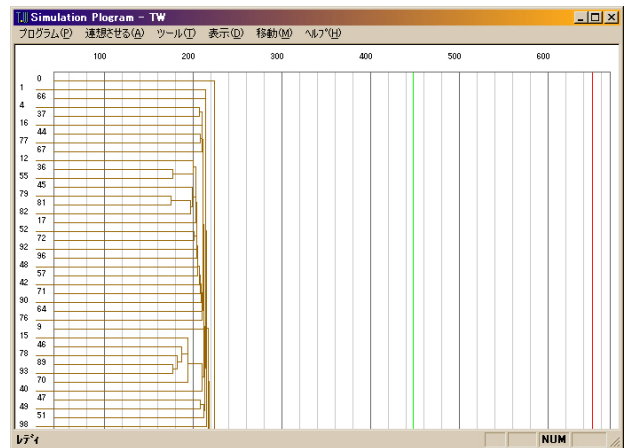
ューを選択することで達成される。



図D. 6 最短距離法によるデンドログラムの表示を選択する



図D. 5 パラメータの設定ダイアログ



図D. 7 最短距離法によるデンドログラムの表示

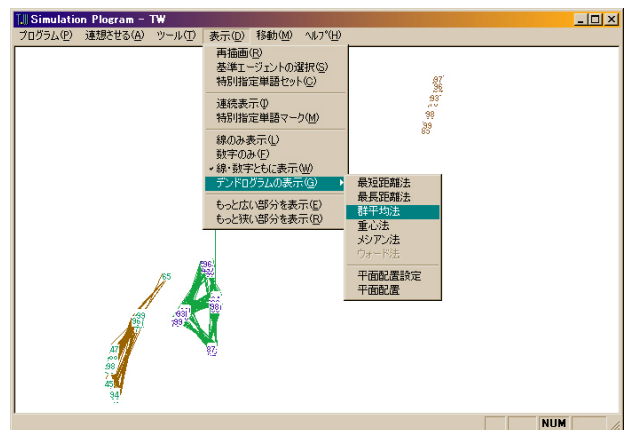
「会話回数を指定」メニューを選択すると、図D. 4のようなダイアログが現れる。ここで、希望するターン回数を指定するようになっている。図D. 4では例えとして200ターン行うよう指定している。

指定を終えて「OK」を押すと会話が始まる。会話途中では左上に図D. 5のようなダイアログが現れて、いつ会話終了するかわかるようになっている。あまり回数が多いと、なかなか終わらないので、最大 50000 回ぐらいを目安として指定する事をお勧めする。

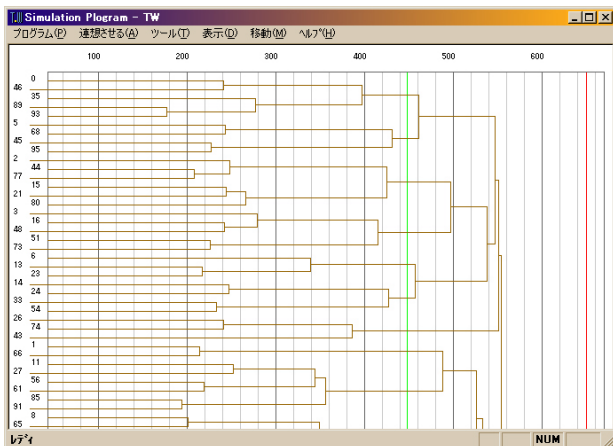
D. 2. 3 デンドログラムの解析

会話が終了すると、エージェントの類推マトリックスがどうなったかを解析するフェーズに移る。そこでクラスタ分析というわけだが、類推マトリックスをクラスタ分析に掛け、その結果をデンドログラムという表示形式で見ると、図D. 6に示すようなメニ

図D. 6、図D. 7の例では最短距離法を通じて分析した結果を表示したものであったが、群平均法を使って分析するときには以下の図D. 8のようにメニューを選択してやると、群平均法を使って分析し、図D. 9にあるようなデンドログラムを表示してくれるようになる。



図D. 8 群平均法によるデンドログラムの表示を選択する



図D. 9 群平均法によるデンドログラムの表示

どの方法によって分析された結果のデンドログラムなのか、こうした表示のみではよく解らないので、それを明示的にウィンドウのどこかで表示さるべきであった。

D. 2. 4 その他

以上で実験手順の解説は終わったが、以下では分析しやすいように付加した TW の機能について、幾つか説明する。

D. 2. 4. 1 移動

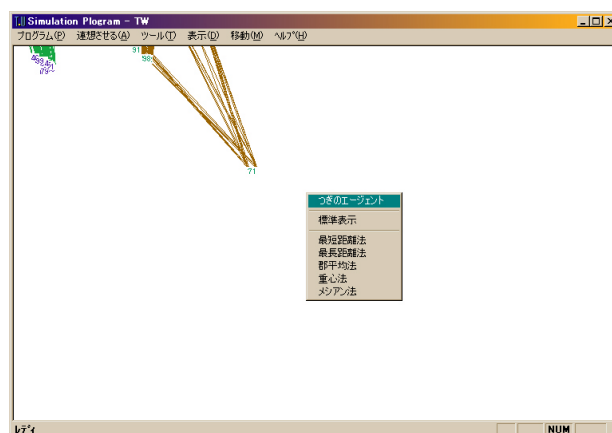


図D. 10 移動させるときのメニュー

デンドログラムを表示する際、表示物の一部がウィンドウからはみ出してしまう事態が多々ある。こんなとき表示されているものを移動させたいが、そんなときは図D. 10にあるようなメニューを使って移動させるなり、キーボードの「↑」「↓」「→」「←」を押して、移動させるかして対処できるようになっている。

また、表示形態を「デンドログラム」にしたり、初期表示形態に戻したりするには図D. 11にあるようなポップアップメニューを右クリックして出し、そのメニューから選択することで可能になっている(図D. 12参照)。それ以外にも、ウィンドウ上部の

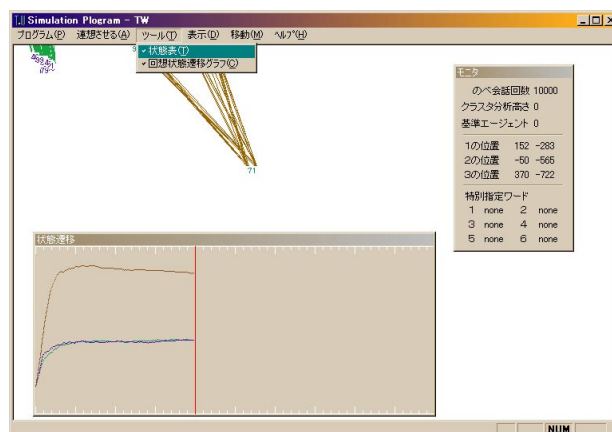
メニューにある「表示」の中のサブメニューから表示の種類を選択することもできる(図D. 14参照)。



図D. 12 ポップアップメニューをだしたところ

メニュー中の「次のエージェント」は、分析対象とするエージェントを逐次的に選択するときに利用するメニュー。「標準表示」は、初期化直後の表示形式を選択するときに使うメニュー。続く5つはメニューに記された方法で現行で指定されているエージェントの類推マトリックスをクラスタ分析し、その結果をデンドログラムとして表示させるようにするメニュー

D. 2. 4. 2 類推可能単語数の変遷グラフ表示



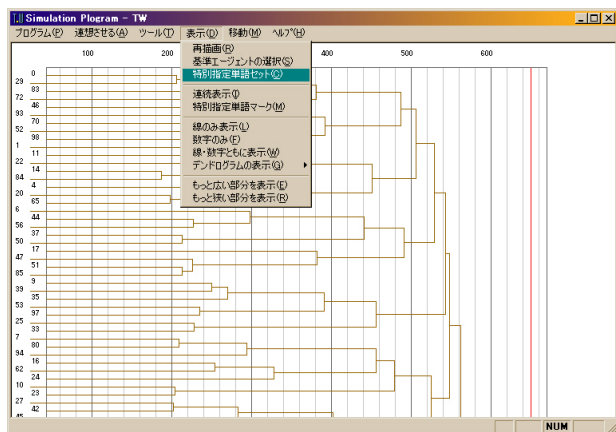
図D. 13 分析ツール

類推可能単語数の遷移グラフを表示するには、メニューの「ツール」にある「回想状態遷移グラフ」にチェックを入れると現れる。遷移グラフのほかにも、ターンの回数をカウントしている値や、フォーカスされている単語(次の節で紹介する)と索引番号との組み合わせなどを表示するモニタウィンドウがあるが、これは「状態表」にチェックを入れることで見られるようになる。

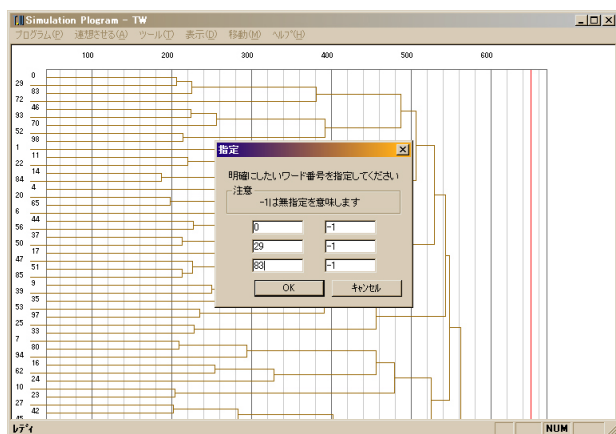
D. 2. 4. 3 ある特定単語をフォーカス

たかだか100語とはいえ、表示させると結構たくさんあるよう

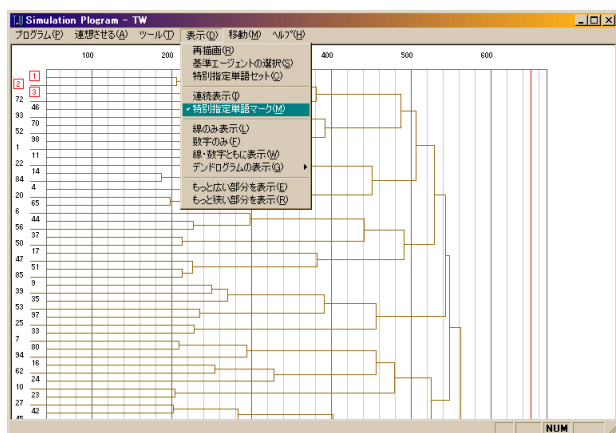
に見える。例えばデンドログラムから目的の単語を見つけるとなると、結構骨が折れる。そこで、目星をつけておきたい単語には、特別に色や番号をつけるなりしておけば見やすくなる。そんなわけで付加したものが、以下の図D. 14から図D. 16で紹介する機能である。



図D. 14 特別指定単語セットのメニューを選ぶ



図D. 15 インデックスをつけたいワード番号を指定する



図D. 16 特別指定単語マークにチェックを入れる

図D. 14、図D. 15で、マークしたい単語を選択する(最大6個まで)。マークした筈の単語を、色つきの口番号として表示するには、図D. 16にあるメニューにチェックを入れる。

今回の場合、デンドログラム上で、上からみつつに相当している「0」「29」「83」を選択した。

D. 2. 4. 4 クラスタの表示(未完成)

クラスター分析をしているので、二次元空間に単語を配置し、デンドログラムのある切り口(結合される距離の軸に対して指定する値)で見たとき、どのようにクラスタが形成されているかを上手に見定めるようにしたい衝動に刺されるわけだが、これがかんがうまくなかった。

仮にもハウスホルダー法と呼ばれる方法で、類推マトリクスから固有ベクトルふたつ(最大固有値と、その次に大きい固有値から得られる固有ベクトル)を導いてこれを座標情報として利用しようとしたが、逆行列を2回求めなければならないがゆえに、大量にメモリを消費する羽目になって、上手く動いてくれななかった(50単語にたいしてでもスワップは150Mbyteを超える。そしてハードディスクの空きが少ない私のマシンは奈落へ)。

ここはひとつ、ハウスホルダー法に比べて計算回数こそ掛ってもメモリを食いそうにないヤコビ法で固有ベクトルを求める方針に変えるべきかもしれないと、今は考えている。

もちろん、今回の場合は非常に要素が多いので、仮に固有ベクトルが上手く求められたとしても、単語達が二次元平面を理想的にばらついてくれるかどうかは、実際にやってみないとわからないのが辛いところである。

D. 2. 4. 5 ほかに

他の機能としては、データの読み書き、パラメータの読み書きがある。これはウインドウ上部にあるメニューの「プログラム」の中に、その一連の読み書きを行う選択肢が収められている。だが、時々妙な挙動を示すので多用することを勧められない。

この読み書き機能はよく使われうる機能であるにもかかわらず、他の機能として控えめに記したのはそのバグが気になったからである。

D. 3 バージョンアップについて

プログラムとしてはさっぱり洗練化されていない。バグも山ほど残っている。今回はバグ取りまでしつこくできなかったが、機会があったらバグ潰しをしたい。

それに、やはり単語を二次元平面に美しく配置させたいところである。是非、固有ベクトルを瞬時に算出できる効率的なア

ルゴリズムを実装し平面における可憐なクラスタ表示を実現したい。