ヒューマンインタフェース 52－2
（1994．1．13）

KJ法グループ編成段階においてカードを自動配置する方法の検討

新田清, 杉山公造
(株) 富士通研究所 情報社会科学研究所

KJ法グループ編成段階は，人が関係ありと思ったカードをグループに分けていく過程である．本稿ではそのような過程に計算機によるカードの自動配置を利用する方法を検討する．われわれはこの方法の中核として働くアルゴリズムの候補としてクラスタ分析に使われる多次元尺度構成法に着目した．そしてKruskalおよびKakushoがそれぞれ提案する写像を実装し挙動を調査することにより，グループ編成段階に用いることが適切であるかどうかを検証した．写像で用いる規準関数の解が非最小極小に陥り最適な配置が得られない可能性があるなどの問題点を整理した．そしてそれら問題のうちのひとつについてその性質を実験を通して明らかにしたことで，グループ編成という目的に対する解決法を見つけることができた．

An investigation of the method for the automatic layout of cards in the grouping phase of the KJ-method

Kiyoshi NITTA and Kozo SUGIYAMA
Institute for Social Information Science,
FUJITSU LABORATORIES LTD.
140 Miyamoto, Numazu-shi, Shizuoka, 410-03 JAPAN

In the grouping phase of the KJ-method, the user gathers and places the cards perceived to be related. In this paper, we investigate the methods by which the computer can automatically place the cards specified by the user. We consider a multidimentional scaling approach which has been used for cluster analysis, in the expection that it would be suitable for an automatic layout algorithm. We implemented and evaluated two multidimentional scaling mappings, proposed by Kruskal and Kakusho. We found that it is possible both the mappings to be trapped in local minima. Thus the computer does not calculate the best layout. While we can clarify the nature of one of those problems by observing the results of the mappings, we found the solution limited to the use in the grouping phase.

1 はじめに

最近，創造的問題解決のための発想法として有名なKJ法[1),2)]を参考とした発想支援ツールの研究開発が盛んとなり，様々なアプローチが試みられている[3)]．われわれの研究グループでは，これまで図解の自動レイアウト機能やフィッシュアイ機能を直接操作環境やアニメーション環境で用いることができる高機能グラフィックユーザインターフェースを持つ発想支援ツールD-ABDUCTORの開発を行なってきた[4),5),6)]．現在，D-ABDUCTORはKJ法(狭義のKJ法)の過程全体を支援することができるが，カードを収集した後のグループ編成段階における計算機支援は他の段階に比べまだ十分でなく，カードの移動作業は全て利用者の操作に任せられており，D-ABDUCTORの特徴のひとつである自動レイアウト機能の利用はなされていなかった．そこで本稿では，グループ編成段階におけるカードの自動配置機能の利用について考察し，いくつかの実験を行なった結果を報告する．

ここでは，グループ編成段階におけるカードの自動配置とは，「ランダムに配置されたカード間に利用者が数個の関係を与えるたびに計算機が自動的に関係で結び付けられたカードを互いに近くに配置し，そのために必要なスペースもうまく空けてくれる」ことを行なうことだとする．本稿では簡便のためこの自動配置を親近度分割と呼ぶことにする．配置がうまくなされたか否かの基準として，

a1：グループ分けの明確な視覚化

a2：メンタルマップの保存(配置前のカードの相対的位置を出来るだけ保存する)

の二つを考える．

グループ編成段階におけるカードの自動配置のためには

M1：KJ法における操作に忠実な方法の利用

M2：グラフ自動描画法[7)]の利用

M3：統計的方法の利用

の3種類が考えられる．M1は最も単純であるが方法の発展性は乏しいと思われる．M2に関しては力指向配置法[8)]などの自動描画法が有望であると思われる．プログラムを既に作成済みであるので別の機会にその結果は報告したい．M3は多次元尺度構成法や多変量解析を利用する方法である．カード間の類似度を連想機能などを用いて計算できる場合などへの応用可能性があり，グループ編成支援のための方法として有望であると考えられる．

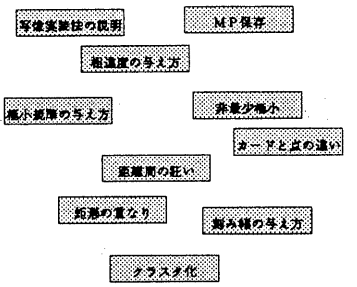

図2: 作業前の状態

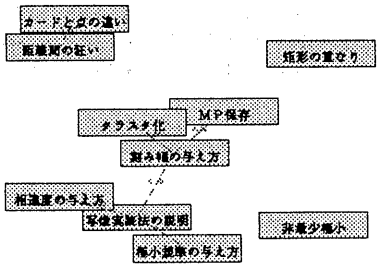

図3: 支援システムを使った作業の途中経過

堀ら[9)]は，多次元尺度構成法を用いた魅力的な発想支援システムを既に開発している．本稿ではこの研究に刺激を受け，多次元尺度構成法であるKruskal[10),11)]やKakusho[12)]の方法をグループ編成段階におけるカードの自動配置に用いることを考察し実験を行なった．

2 距離順序保持写像とその実装

Kruskalの方法やKakushoの方法はそれぞれ特徴のある写像として定義される．両写像に共通する性質として「任意のデータについて他のデータを多次元での距離[1]で順序付けたときに，写像後の低次元においてもその順序をなるべく保存しようとする」性質があるため，ここではそれらを距離順序保持写像と呼ぶことにする．

われわれは実験のため距離順序保持写像を実装した．図2, 3にその作業の様子を示す．KJ法グループ編成段階を始めるときの状態ではカードはランダムに並べられる(図2)．そして距離順序保持写像を対話的に呼び出しながら図3のような配置を得る．

[1] 本節ではこれをユークリッド距離とする．ただしKruskalにおいてはそれに限定されない．

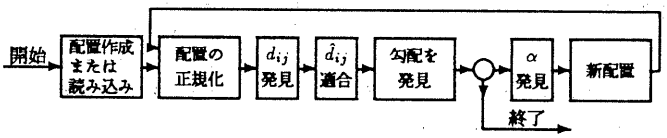

図 1: Kruskal の写像のブロックダイアグラム

2.1 Kruskal の写像

概要

Kruskal が文献[10),11)]で提案した写像を本稿では**Kruskal の写像**と呼ぶことにする．Kruskal の写像は多次元空間のデータをそれより低い次元に写す写像である．多次元空間のデータからそれらデータ間の相違度を求め，その相違度を低い次元の配置に反映させる．その反映の度合を測るものとして *stress* と呼ぶ規準関数を導入している．stress は相違度と低次元配置の歪みを表すよう意図されている．Kruskal の写像は stress を最小化する勾配法として実現される．

まず stress について説明する．n 個のデータ $1, \ldots, n$ があり，任意のデータ $1 \leq i, j \leq n$ の間に相違度 δ_{ij} が定まっているとする．これらのデータが t 次元空間に配置されているとき，stress S は

$$S = \sqrt{\frac{\sum_{i,j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i,j} d_{ij}^2}} \qquad (1)$$

となる．ここで d_{ij} はその配置におけるデータ i とデータ j の間のユークリッド距離である．$\hat{d}_{ij}$ は δ_{ij} と同じ順序を持ち S を最小にするような値を持つ．

Kruskal の写像はこの S に関する勾配法によって実現される．処理の概要をブロック・ダイアグラムを示すと図 1のようになる[2]．

実装について

われわれは Kruskal の写像を実装した．Sun ワークステーション (SunOS 4.1.2) 上で動作しており，c++ (実際は g++2.4.5 で動作確認) で記述している．ソースは約 1,000 行である．c++ 用のフリーのウィンドウライブラリである InterViews を用いて X11R5 上で動作可能な対話インターフェースも実現した．ここでは Kruskal の写像を実装する際に留意した点を列挙する．

[2] 図 1は文献[11)]に掲載されている図を翻訳したもの．その文献では各部分について詳細な説明がなされている．

相違度　KJ 法グループ編成段階から離れると，カードどうしに相違度を与える方法としては様々なものが考えられる．まず与える形態としては二値で与える方法と多値で与える方法が考えられる．二値で与える方法では関係のあるカードどうしの相違度は 0，ないものは 1 というように与える．多値で与える方法では関係のないものは 5，少し関係のあるものは 3，そして密接な関係のあるものは 1 というように与える．また何をもとに相違度を与えるかについては利用者の判断を使用する方法や，カードのもつ (キーワードなどの) 属性を利用する方法が考えられる．

本実装ではこれらの異なった実験を可能にするため，多値の相違度を利用者が自由に与えられるようにした．ただし本稿では KJ 法グループ編成段階の支援が主目標であるので，今回の実験では二値の相違度を利用者の判断により与えている．

刻み幅の初期値　刻み幅の初期値は慎重に決定しなくてはならない．一般に勾配法においては刻み幅の初期値 α が大きいと初期配置からの変動も大きい．よってあまり無駄なループを繰り返さずに結果を得ることができる．また後で述べる非最小極小を回避しやすくなる．しかしメンタルマップを保存する (達成規準 a2) ためには α は小さい方がよい．

よって本実装ではこれらのバランスをとりながら適切な α を選択できるように留意した．

反復終了条件　Kruskal の写像を構成するループの反復終了条件は二つある; 1) stress が 0 である，2) 勾配の相対規模が十分小さい．stress が 0 にならない場合には条件 2) で終了を判定することになる．ところがこの判定法は文献[11)]にも指摘しているとおりデータの統計的変動に影響を受ける．つまりデータの数などに応じて判断規準が変化する．親近度分割においてはデータ (カード) の数をあらかじめ想定することはできない．よってそのような場合でもループが終了するように，「与えられた最大ループ回数を越えない」という条件を加えた．

2.2 Kakusho の写像

概要

文献[12]で述べられている写像を **Kakusho の写像**と呼ぶことにする．Kakusho の写像は Kruskal の写像を拡張したものである．stress の代わりに k-stress という規準関数を用いる．k-stress は局所的なデータの相違度を反映させるよう意図されている．どの程度局所的かの度合をパラメタ k により与えることができる．

k-stress について説明する．n 個のデータ $1, \ldots, n$ があり，それらの間に相違度 δ_{ij} が定まっているとする．その相違度をもとにデータ i に対して k 番目以内に近いデータのリスト[3]を考え，それを $\Omega_k(i)$ と表す．これらのデータが t 次元空間に配置されているとき，k-stress S_k は次のように定義される．

$$S_k = \sqrt{\sum_{i=1}^{n} \frac{\sum_{j \in \Omega'_k(i)} (d_{ij} - \hat{d}_{ij})^2}{\sum_{j \in \Omega'_k(i)} d_{ij}^2}} \qquad (2)$$

ここで d_{ij} はユークリッド距離である．$\Omega'_k(i)$ はその配置での距離をもとにしたデータ i に対する近い順のデータのリストであり，かつ $\Omega_k(i)$ を含む最小のリスト[4]である．$\hat{d}_{ij}$ は $\Omega_k(i)$ と同じ順序を持ち S_k を最小にするような値を持つ．

Kakusho の写像も Kruskal の写像と同様のループからなる．ただし k-stress 計算のために $\Omega'_k(i)$ の計算が新たに必要になる．また stress の定義が異なるためその勾配も Kruskal のものとは違っている．勾配については文献[12]で公式が与えられている．

実装について

Kakusho の写像も c++ で実装した．c++ では Kruskal の写像の派生クラスとして Kakusho の写像を書いた．ソースの追加分は約 250 行となった．その他の実装に関する留意点は Kruskal の写像と同様である．以下，Kakusho の写像についてとくに問題となった点を列挙する．

$\hat{d}_{ij}$ の拡張　文献[12]においては $\hat{d}_{ij}$ が厳密にいうと完全には定義されていない．k-stress の定義によると $\hat{d}_{ij}$ は任意の $j \in \Omega'_k(i)$ に対して定義されていなくてはならない．しかし文献[12]では $\hat{d}_{ij}$ は $\Omega_k(i)$ と同じ順序を持つとしか説明されていない．一般に $\Omega_k(i) \subseteq \Omega'_k(i)$ である．よって $R = \Omega'_k(i) \setminus \Omega_k(i)$ としたとき $j \in R$ に対する $\hat{d}_{ij}$ は未定義となる．

これへの対処法は二つ考えられる．

1. $\hat{d}_{ij}$ は $j \in \Omega_k(i)$ で定義されていればよいとする．
2. $\forall \ell \in \Omega_k(i)$ に対して $j \in R$ は ℓ よりも後の順序を持つとし，$\hat{d}_{ij}$ を求める．

1. をとった場合，k-stress が「不必要なデータが近接してもそれを考慮しない」性質を持つことになる．これはわれわれの目的から見て都合が悪い．また文献[12]でもこれを意図しない．よって実装時の対処法としては 2. を採用した．

ループ毎の $\Omega_k(i)$ の変更　本来 $\Omega_k(i)$ は Kakusho の写像内のループを通して不変である．ところがわれわれの与える相違度 δ_{ij} は二値である．このとき $\Omega_k(i)$ はループ毎に更新されることが望ましい．この理由を以下の例で説明する．

データ 1 に対して相違度 0 のデータが $\{2,3,4\}$，相違度 1 のデータが $\{5,6,7\}$ であるとする．このときに $\Omega_4(i) = \{2,3,4,5\}$ とするのは正しい．しかしこれは $\{2,3,4,6\}$ でも $\{2,3,4,7\}$ でも正しい．つまり $\Omega_k(i)$ の選択には自由度が存在する．たまたま決定した $\{2,3,4,5\}$ で，その中に 6,7 でなく 5 が入っていることにとくに意味はない．

$\Omega_k(i)$ を固定した場合，ループを繰り返す過程でその意味のない選択を配置に反映しようという力が働く．これは意図しない配置の移動を生ずることになる．これはわれわれの目的から見て都合が悪い．よってループ毎に $\Omega_k(i)$ をその自由度の範囲内で現在の配置を反映するよう実装した．

3 実験結果と考察

この節では Kruskal と Kakusho それぞれの写像を実装して調べた挙動について述べる．利点・問題点・その解決法をそれぞれまとめて述べる．両写像の挙動は非常によく似ている．以下のほとんどがそれらに共通して言えることである．

3.1 利点

相違度の小さいカードを近くに配置する

これは親近度分割の達成規準 a1 の重要な性質のひとつである (第 1 節)．調査した両写像ともこの性質を満たしている．

実装した写像による配置例を図 4, 5 に示す．この図では見やすさのためカードを点で表した．カー

[3] 例えばデータ 1 に関して $\delta_{12} = 0.2, \delta_{13} = 0.5, \delta_{14} = 0.9, \delta_{15} = 0.3$, であったとすると，3 番目以内の近いデータのリストは，$\{2,5,3\}$ となる．

[4] 例えば $\Omega_3(1) = \{2,3,4\}$ で，そのときの配置でのデータ 1 に対する近い順のデータのリストが $\{2,3,5,4,6,\ldots\}$ であるとき，$\Omega'_3(1) = \{2,3,5,4\}$ となる．

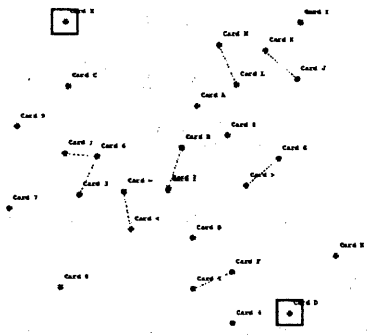

図 4: 利用者による関係の設定

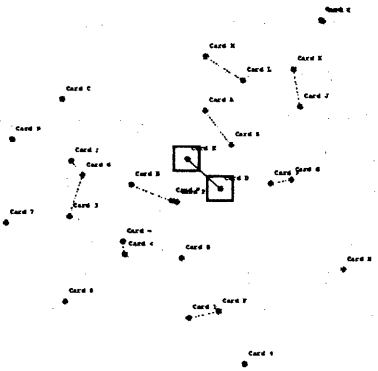

図 5: Kruskal 写像による親近度分割

ドをつなぐ線分はそれらのカードの相違度が小さい (親しい) ことを示す. 図 4 は Kruskal の写像に入力する初期配置を示す. 利用者はこの配置を見て正方形で囲んだカードを近いと考え, それらに親近度を与えたとする. この正方形は見やすさのため写像出力に後から付け加えた. 図 5 はその出力の配置結果である. これからわかるように親近度を与えたカードは近くに配置される.

メンタルマップを保存する

これは親近度分割の達成規準 a2 である (第 1 節). 調査した両写像とも a2 を満たす傾向にある.

ここでメンタルマップとは親近度分割作業中に人間が視覚情報として一時的に記憶しているカードなどの配置を指している. 自動レイアウトの前の配置と後の配置があまり違わなければメンタルマップはほぼ維持される. このとき人間はその維持されたメンタルマップの作用により, 初めてその配置を見るときよりも速やかに多くのカードの中から目的のカードを発見することが可能になると考えられる.

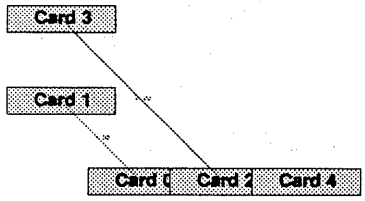

図 6: 初期配置

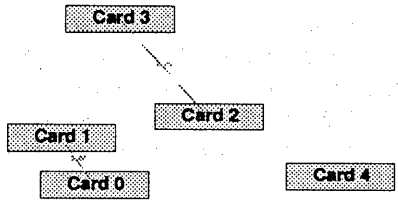

図 7: 初期配置からの変化が少ない例

実際の KJ 法グループ編成段階における, 計算機を使わない親近度分割ではメンタルマップはほぼ保存されている. このときの親近度分割ではカード全体の移動量は小さい. なぜならば人間がカードを手で動かすので一度に大量のカードを移動させることが困難だからである.

距離順序保持写像がメンタルマップを保存する理由は以下のように説明できる. 距離順序保持写像のループは stress が 0 または十分小さくなれば終了する. 一般にそのような配置は数多く存在する. 距離順序保持写像は初期配置を少しづつずらしながらそのような stress の小さい配置にもっていく. ところで親近度分割で距離順序保持写像を使うときには相違度の変化は少ない. 既存の配置を生成した時点から比べると 2, 3 のデータ間の相違度が変化するくらいである. つまり距離順序保持写像への初期配置となる既存の配置はもともと stress が小さいことが予想できる. そのため相違度が変化したデータを中心に移動を行なうだけで stress が十分小さくなる. よって全体の配置が初期配置から大きく変わることはない. これは図 4, 5 からもわかる.

また図 6, 7 から, 相違度が変化したデータについても初期配置からあまり変わらない配置が得られていることがわかる.

ただしこの性質は刻み幅の初期値 α があまり大きくない場合に顕著に現れる. 刻み幅は距離順序保持写像内のループが回る毎に変化する. stress の勾配

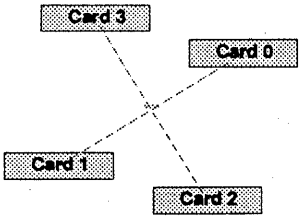

図 8: 非最小極小 (stress=10.8%)

が小さくなれば刻み幅も小さくなる．多くの場合，刻み幅は単調減少である．よって刻み幅の初期値 α が初期配置からの変動の大きさを決定すると考えてよい．α があまり大きいと距離順序保持写像内の初回のループで勾配方向に動き過ぎる．この時は初期配置とはかなり違った配置が得られてしまう．

3.2 問題点

非最小極小

勾配法の限界としてよく知られていることだが，ループをいくら繰り返しても規準関数の最小ではなく極小に陥って抜け出せないことがある．例えば図 8 のような配置になったとき距離順序保持写像の解は非最小極小に陥っている．

stress の非最小極小に陥いることは文献[11]でも指摘している．そこでは解決法として次のものを挙げている．ループの終了条件に達したとき，刻み幅の初期値 α を極端に大きくしもう一度終了条件に達するまで繰り返す．ランダムな初期配置からやり直してもよい．これを数回繰り返し stress が小さい配置を選ぶ．

ただしこれを採用するとメンタルマップ保存という利点を失う．この利点を保ったまま非最小極小を解決する方法の出現が待たれる．

これとは別に Kakusho の写像においてはパラメタ k を小さくする[5]ことで非最小極小に陥いり難くすることができる．しかしこの場合 k-stress の定義からも明らかなことだが，あまり k を小さくするとクラスタが生成され難くなる．例えば $k=1$ とすると stress はたいてい 0 になるが，親近関係が二つ以上あるデータの最も親しいもの以外の関係は全て stress 計算中無視される．これは親しいものを近く，そうでないものを遠くという基本的な距離順序保持写像の性質を半ば失っている．よってこれは解決策とはなりえない．

[5] データに対する近いもの順のリスト $\Omega_k(i)$ の大きさを減らす．

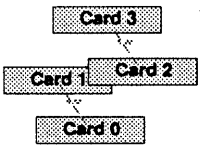

図 9: 点の距離がうまく感じとれない例

遠近の区別

数値上では遠近の区別があってもそれが人間には感じられないことがある．例えば図 7 において矩形重心間の距離を測ると，Card 2 と Card 3 の間は 1.23 であり，Card 2 と Card 4 の間は 1.30 である．確かに Card 3 は Card 4 よりも Card 2 に近いが，それを人間が感じとるのは困難である．

矩形と点の距離感の違い

本稿でとりあげた二つの距離順序保持写像は配置するデータを点として扱っている．このときデータどうしはどれだけ近くに配置されても構わない．データの間の距離は点と点の間の距離である．しかし実際にグループ編成支援で計算機に処理をさせたいのは矩形のカードである．Kruskal の写像，Kakusho の写像のいずれもその出力する配置において矩形が重なってしまうことが多い．この重なりの問題は本写像を実装する前から予想していた．それゆえ親近度分割からは矩形の重なりの処理を外したが，これは今後解決すべき問題である．

また実際には近いカードが遠いように感じて見えることもある．図 9 において，Card 1 と Card 0 の重心の間の距離は Card 1 と Card 2 の重心の間の距離よりも小さい．しかしこの図を見た場合 Card 1 と Card 2 の方が近く見える．つまり矩形と点では人間が感じる距離感に差が存在している．

3.3 問題点の解決

今後の方向を考えたとき，まず距離順序保持写像を使うかどうかの選択肢がある．本稿ではこれを使うことにして話を進める．

距離順序保持写像を使う場合は前小節で述べた問題点を解決しなくてはならない．われわれは実装した距離順序保持写像を使った実験を通していくつかの問題に関しては解決の見通しを得た．また別のいくつかの問題に関しては解決すべき問題点を掘り下げた．ただし現在のところそれらの実装は行なっていない．

これらの問題がすべて解消されれば親近度分割に距離順序保持写像は適切であると言えるだろう．

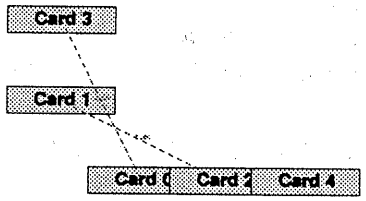

図 10: 非最小極小を生み出す初期配置

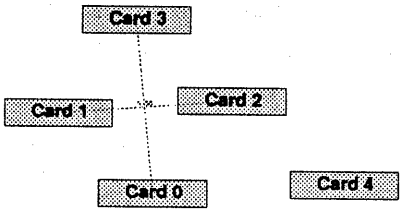

図 11: 刻み幅初期値 $\alpha = 0.2$ の場合

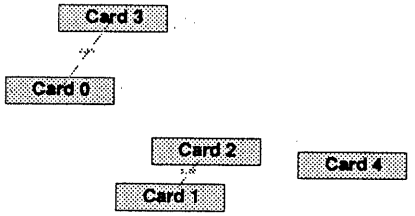

図 12: 刻み幅初期値 $\alpha = 0.6$ の場合

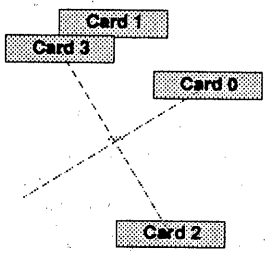

図 13: 非最小極小を解消する移動

非最小極小

刻み幅初期値の調整　刻み幅初期値 α を大きくするとメンタルマップを保存する性質が失われることは前述した．しかしながらこれを大きくすることにより非最小極小を解消できる場合がある．

例えば図 10 を初期配置として Kruskal 写像を適用するとき，刻み幅初期値を 0.2 とすれば図 11 (このときの stress = 10.2%) のような非最小極小配置を，0.6 とすれば図 12 (このときの stress = 0%) のような最小配置を出力する．この場合，刻み幅初期値を 3 倍にすることで非最小極小が回避できたことになる．

非最小極小を配置構造をもとに解消する　親近度分割で起こる可能性がある stress の非最小極小には特徴がある．これは今回の実験における相違度の与え方に依存している．二値の相違度のうち値の低い (親しい) データの間に線を引くとする．このようにして親近度分割でよく起こる非最小極小の配置を観察すると，それらの線が交差していることがわかる (図 8 参照)．よってこの交差線を解消する初期配置を与えることができれば非最小極小も解消できる．

例えば図 8 のような配置が出力として得られた場合，Card 1 を Card 3 の少し右上に動かし (図 13)，これを Kruskal の写像の初期配置とすると，図 14 のような最小配置が得られる．これら 4 つのカードの配置は変わるがそれ以外のカードの位置はほとんど変化しない．よって α の調整よりもこの方法は比較的よくメンタルマップを保存する．

一般に stress が非最小極小に陥っているとき，いつも上の例のような単純な形をしているわけではない．しかし親近度分割においては新規に付け加わる関係の数はほとんどの場合 1 または 2 である．よって上の例で示したような単純な形で非最小極小の原因となる箇所を見つけることができる．

その他の問題

上で解決法を示した問題以外にも，遠近の区別，重なりの解消，そして距離感の違いの解消などの問題があった．これらについては次のような解決の方針を設定している．重なりの解消については，メンタルマップを保存したままカード矩形の重なりを解消する写像を別に用意し，それを距離順序保持写像と組み合わせて使うことを考えている．これは既にいくつかの手法が考案されている [13]．距離感の違いの解消については，矩形の間の距離を新たに定義してそれを元に stress 勾配を求め直すことを考えている．

しかし現在の実装においても利用者の操作を前提とすれば，これらの問題の解決はできる．つまり遠近感や重なりそして距離感の食い違いを利用者自身によるカードの移動によって解消するのである．これを少しでも容易にするために関係のあるカード間に線分を表示するなどの計算機側の支援を実現した．

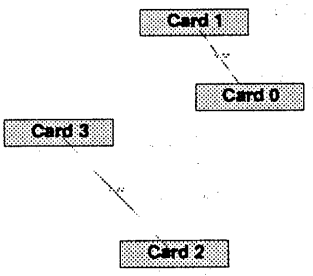

図 14: 非最小極小が解消された例

4 おわりに

人間の創造的思考支援の一要素として KJ 法グループ編成段階の親近度分割作業に注目した．親近度分割を計算機で実現する手法として Kruskal の方法や Kakusho の方法が利用できるかどうかを判断するためにそれらを実装して調査を行なった．その結果，目標として設定した達成規準を不十分ながら達成することが確認できた．ただし規準 a1 に関しては，配置の歪みを表現する規準関数が非最小極小に陥ったり，データがカードとして表現されたときに重なりが生じる，そして得られたカードの配置が必ずしも人間の遠近感と一致しないなどの問題点がある．これら問題のうち非最小極小の問題について，グループ編成に限った場合ではあるが，規準 a2 を損なわない一解決法を見つけた．今後残りの問題を解決しなくてはならない．

謝辞

本テーマ全般にわたって熱心に議論していただいた同研究所の三末和男氏に感謝します．

参 考 文 献

1) 川喜田二郎：発想法, 中公新書 (1967).

2) 川喜田二郎：続・発想法, 中公新書 (1970).

3) 杉山公造：収束的思考支援ツールの研究開発動向 — ＫＪ法を参考とした支援を中心として，人工知能学会誌, Vol. 8, No. 5, pp. 568–574 (1993).

4) 三末和男，杉山公造：図的思考支援を目的とした複合グラフの階層的描画法について，情報処理学会論文誌, Vol. 30, No. 10, pp. 1324–1334 (1989).

5) 三末和男，杉山公造：図的発想支援を目的とした図の多視点遠近画法について，情報処理学会論文誌, Vol. 32, No. 8, pp. 997–1005 (1991).

6) 三末和男，杉山公造：図を対話メディアとする発想支援システム D-ABDUCTOR, 情報学シンポジウム 講演論文集 (1994), 発表予定.

7) 杉山公造：グラフ自動描画法とその応用，計測自動制御学会 (1993).

8) Fruchiterman, T. M. J. and Reingold, E. M.: Graph Drawing by Force-directed Placement, *Software - Practice and Experience*, Vol. 21, No. 11, pp. 1129–1164 (1991).

9) Hori, K. and Ohsuga, S.: Assisting the Articulation of the Nebulous Mental World – Toward Computer Aided Creation –, 電子情報通信学会第二種研究会「言語獲得・概念形成」LA91-5 (1991).

10) Kruskal, J. B.: Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis, *PSYCHOMETRIKA*, Vol. 29, No. 1, pp. 1–26 (1964).

11) Kruskal, J. B.: Nonmetric Multidimensional Scaling: a Numerical Method, *PSYCHOMETRIKA*, Vol. 29, No. 2, pp. 115–129 (1964).

12) Kakusho, O. and Mizoguchi, R.: A New Algorithm for Non-linear Mapping with Applications to do Dimension and Cluster Analyses, *Pattern Recognition*, Vol. 16, No. 1, pp. 109–117 (1983).

13) Eades, P., Lai, W., Misue, K. and Sugiyama, K.: Preserving the Mental Map of a Diagram, in *Proc. of the First International Conf. on Computational Graphics and Visualization Techniques(COMPUGRAPHICS '91)*, pp. 34–43 (1991).