知識工学と人工知能　45－1
（1986・3・10）

２次元図形のレイアウト問題を解くＰｒｏｌｏｇ型言語

加藤　誠巳　　藤森　福夫
（上智大学理工学部）

１．まえがき

決まった形状を持った２次元図形の空間配置問題を計算機によって解かせようとする試みは従来から行なわれている〔1〕。例えば住宅間取り設計、ＬＳＩのマスク設計などはこの２次元図形ユニットの配置問題として扱うことができる。そのため制限された配置空間内に２次元図形ユニットを所定のルールを満たすように隙間なく配置し、その総ての解を重複することなく見い出す問題を解くプログラミング言語の開発が望まれる。

本稿では配置するユニットと配置平面内の空所の２次元パターン・マッチング、配置されたユニットの状態ならびにユニット間の関連条件のルール・パターン・マッチング、及びいずれかのパターン・マッチングが失敗した際に自動バックトラックする機能を有する論理型言語による解探索手法について検討を行なったので、その結果について御報告する。

２．２次元図形ユニットの形状および配置空間の表現

配置する２次元図形ユニットは同一の正方形のセルで構成できるものに限定する。従って単位セルとして配置すべき総ての２次元ユニットを構成できる正方形のうち最も大きな正方形を選べば２次元ユニットを構成するのに必要なセル数は最小となる。一方配置空間もこの単位セルで構成されるものに限定する。配置空間には与えられた２次元ユニットを隙間なくつめるので配置空間を構成する単位セルの数は２次元ユニットを構成する単位セルの総和に等しい。

単位セルを正方形としたので各２次元ユニットの配置モードは90度回転と反転により最大 8通りあるが、解探索実行時には入力パラメータにより、使用する配置モードを制限することも出来るようになっている。また配置空間は東西南北の方向を有している。

３．２次元ユニット配置問題の解探索アルゴリズム

制限された配置空間に隙間なく２次元ユニットを配置する問題の総ての解を求める手法は多数存在するが、次の２つの手法に大きく分けることができる。ここで配置空間のセルの番号付けは図１に示すようなラスタ走査式を用いるものとする。（この場合には縦方向に番号付けを行なったが横方向でもよい。）

(1) 配置するユニットの順序を決定し、各ユニットを置く配置空間の位置を順次変えて配置する手法　（若番ユニット優先）
(2) ユニットを置く配置空間の位置の順序を決定し、配置するユニットを順次変えて配置する手法　　（若番空所優先）

この他に若番ユニット優先と若番空所優先の方法を組み合わせたものも考えられる。若番ユニット優先の方法を用いた場合に生じる問題点を図１、図２のモデルを用いて説明する。　このモデルでは図３のように配置空間の 1,3,5,13,15,17,25,27,29の 9つの位置に２次元ユニットの左上隅が配置されたときのみ解となる。従って解総数は異なった 9つの位置に異なった 9つのユニットを割り当てることになるので 9!=362880である。

ところで、若番ユニット優先の方法を用いたとき１手目にユニットを配置できる位置は 25 ある。この時点で 25-9=16もの無意味な探索枝が生じる。(図４) ２手目にユニットの左上隅を配置できる位置は図５に示されているように１手目にユニットを置いた位置により数が異なる。図中の”通り数”は左右上下の対称性を考慮した場合の数を表わしている。この時点で生じる無意味な探索枝は 13x4+19x8+11x4+16x4+16x4+8x1=384 となる。このような無意味な探索枝が解探索時に起こるバックトラックを増加させる要因となる。

一方、若番空所優先の方法を用いたときは１手目で生じる探索枝はユニット総数に等しくこの場合は９である。２手目に配置を試みるユニットの数は、１手目に配置したユニットを除く８である。同様に最後の手の９手目まで配置を行なったときに生じる枝の数は9!=362880 となり解総数と同じで一度もバックトラックは生じないため極めて探索効率が良い。（この場合には効率は100%である。）

以上述べた例は若番ユニット優先と若番空所優先の方法の差が極端な例であるが、一般に同様なことが言えるので、ここでは若番空所優先の方法を採用することにする。図６のモデルに対し横方向ラスタ走査番号付けを行ない、若番空所優先の方法を用いた解探索の様子を図７、図８に示す。

1	7	13	19	25	31
2	8	14	20	26	32
3	9	15	21	27	33
4	10	16	22	28	34
5	11	17	23	29	35
6	12	18	24	30	36

図１　配置空間（モデル１）

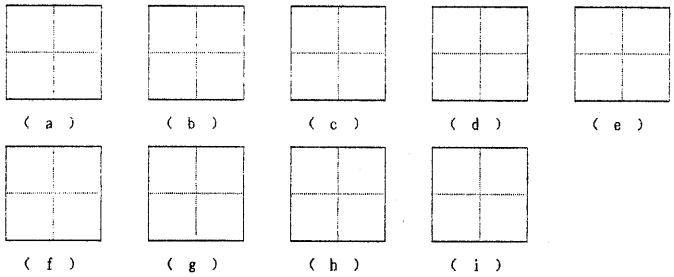


図２　２次元ユニット（モデル１）

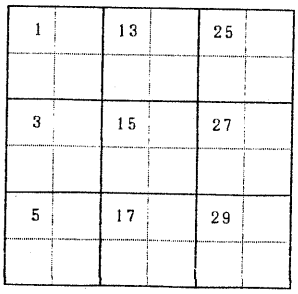


図３　モデル１の解

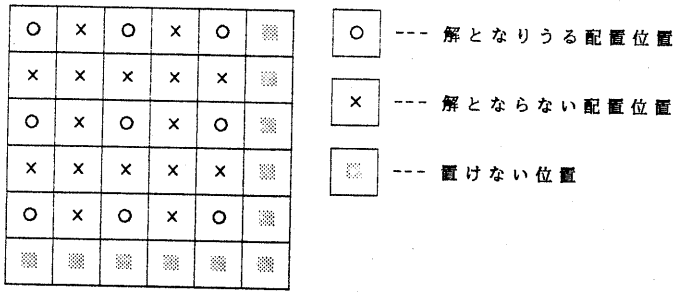


図４　１手目にユニットの左上隅を配置できる位置

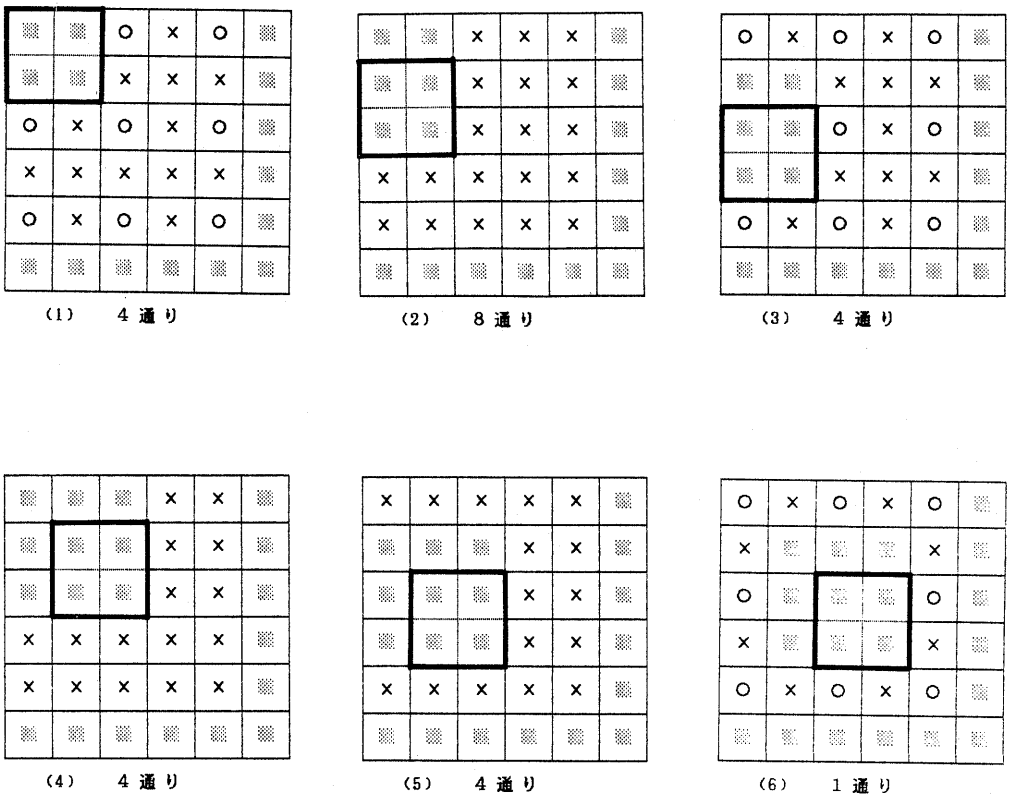


図５　２手目にユニットの左上隅を配置できる位置

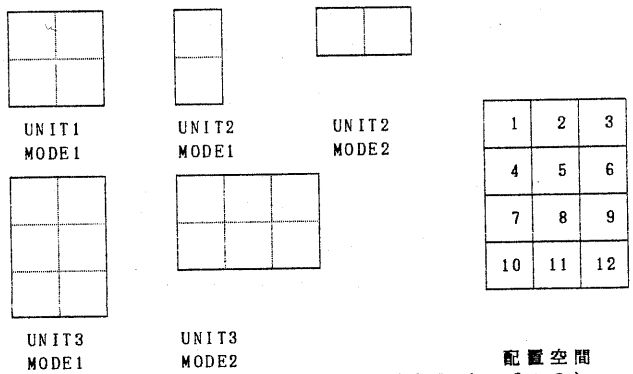
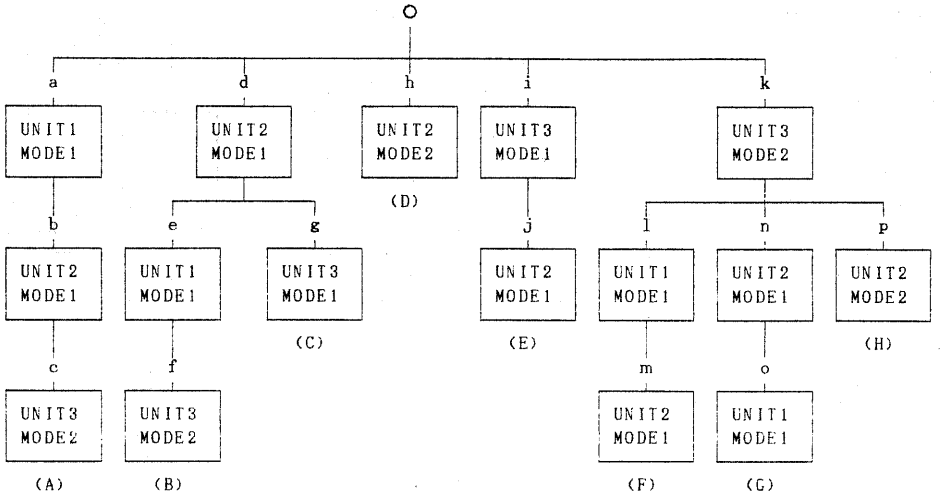


図６　２次元ユニットおよび配置空間（モデル２）



探索順序

○ → a → b → c → d → e → f → g → h → i → j → k → l → m → n → o → p → ○

図 7　解探索枝と探索順序

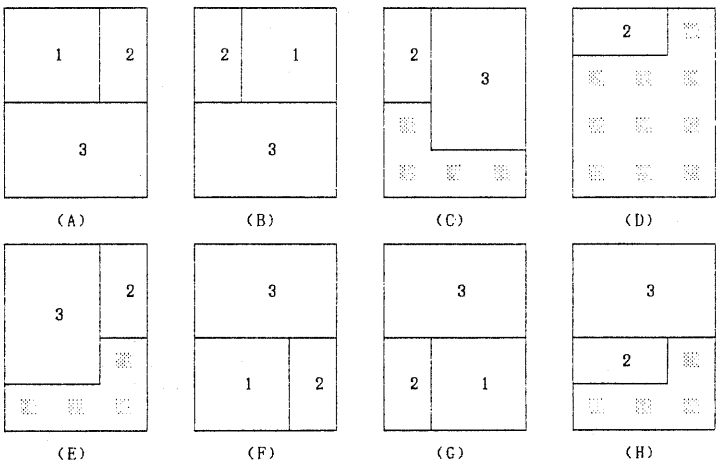


図 8　枝の末端における配置状態（モデル２）

４．２次元ユニット間のルールのデータ構造とルールの探索手法

複数個の２次元ユニット間のルールは方向を有するネットワーク構造として表現することができる。即ち各ユニット自身に関するルールもしくは２つのユニット間のルールを節（ノード）で表わし、この節の親と子の関係をＡＮＤ、兄弟の関係をＯＲに対応させる。節に相応するルールを表現するパラメータを次に示す。但しここでルールに関係する２つのユニットはユニットｉおよびユニットｊであるものとする。

(1) ユニットｉのユニット番号〔1,2,3,...〕
(2) ユニットｉのユニットｊに面する位置［無指定，上，下，左，右］
(3) ユニットｉのユニットｊに面する方向［無指定，東，西，南，北］
(4) ユニットｊのユニット番号〔0,1,2,3,...〕 --- 0は配置空間の外郭線
(5) ユニットｊのユニットｉに面する位置［無指定，上，下，左，右］
(6) ユニットｉとユニットｊが面さなければならない最小単位長　ｎ〔0,1,2,3,...〕
(7) ユニットｉとユニットｊが離れてよい最大単位長　ｍ〔0,1,2,3,...〕
(8) (1) ～ (7)で設定したルールを否定したものをルールとするためのフラグ
〔TRUE,FALSE〕　　但しｎ＝0 のときｍは無効

ここで上，下，左，右とは登録時の形状を基準としており、”面する”という表現はｍ＝0 とすると”隣接する”ということと等価である。また”透明である”という属性を有さないユニットがユニットｉとユニットｊの間に置かれたときは遮られた部分は面していないと見なす。次に節に相応するルールの例を幾つか与える。

＊ユニット１は２単位長以上外郭線と隣接する。
＊ユニット２は下の辺が東を向く。
＊ユニット１は４単位長以下の距離だけ離れて、３単位長以上ユニット３に面する。
＊ユニット２は右の辺が西を向いて西側で２単位長以下の距離だけ離れて、３単位長以上ユニット３の下側に面する。
＊ユニット１は３単位長以上ユニット２と隣接しない。

３個以上のユニット間のルールは前述の節に相応するルールの組み合わせで表現することが出来る。例えば”ユニット１は、ユニット３もしくはユニット４と隣接する。そしてユニット１がユニット２と隣接するときはユニット３とユニット４は隣接せず、またユニット１がユニット２と隣接しないときはユニット３はユニット４と隣接する。”という４つのユニット間のルールは以下に示すＡ～Ｆの節に相応するルールと式(1) に相応するルールのネットワーク構造で表現できる。

Ａ：ユニット１はユニット２と隣接する。
Ｂ：ユニット１はユニット３と隣接する。
Ｃ：ユニット１はユニット４と隣接する。
Ｄ：ユニット３はユニット４と隣接する。
Ｅ：ユニット１はユニット２と隣接しない。
Ｆ：ユニット３はユニット４と隣接しない。
（Ｂ＋Ｃ）・（Ａ・Ｆ＋Ｅ・Ｄ）　　---------- (1)

式(1) に相応するルールのネットワーク構造を図９に示す。

図10に示す２次元ユニットならびに配置空間に対しこのルールを適用したときの解探索途中の幾つかの配置状態（図11）におけるルールの探索判定処理の様子を次に示す。

(a)　○→B〔 0〕→A〔 0〕→F〔 0〕→◎　　総合判定〔 0〕
(b)　○→B〔-1〕→C〔-1〕→○　　総合判定〔-1〕
(c)　○→B〔 0〕→A〔 1〕→F〔 0〕→◎　　総合判定〔 0〕
(d)　○→B〔 1〕→A〔 1〕→F〔-1〕→E〔-1〕→C〔-1〕→○　　総合判定〔-1〕
(e)　○→B〔 1〕→A〔-1〕→E〔 1〕→D〔 1〕→◎　　総合判定〔 1〕
(f)　○→B〔 1〕→A〔 1〕→F〔-1〕→E〔-1〕→
　　C〔 1〕→A〔 1〕→F〔-1〕→E〔-1〕→○　　総合判定〔-1〕

ルールの判定フラグの値の意味　〔 1〕：”ルールを満たしている”
〔 0〕：”ルールの判定ができない”
〔-1〕：”ルールを満たさない”

この場合ルール判定フラグとこの判定値の内容が有効となったユニットの配置段階の情報により、２度以上同じ節に相応するルールの判定処理は行なわないようにしている。従って各配置状態におけるルール判定処理回数の最大値はルールデータのネットワーク構造の節の数に等しい。ルールの探索判定処理の結果、出口（◎）から抜け出たときは次のユニットの配置に進み、入口（○）に戻って来たときはバックトラックし直前に配置したユニットを取り除き他の可能性を調べることになる。

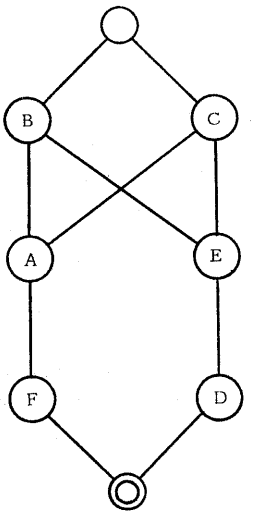


図９　ルールのネットワーク構造

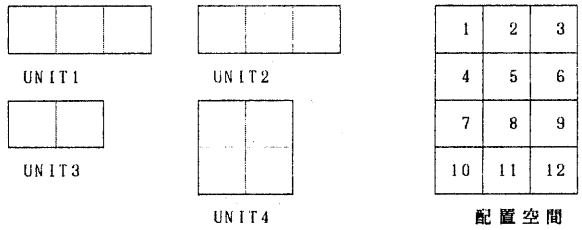


図10　２次元ユニットおよび配置空間（モデル３）

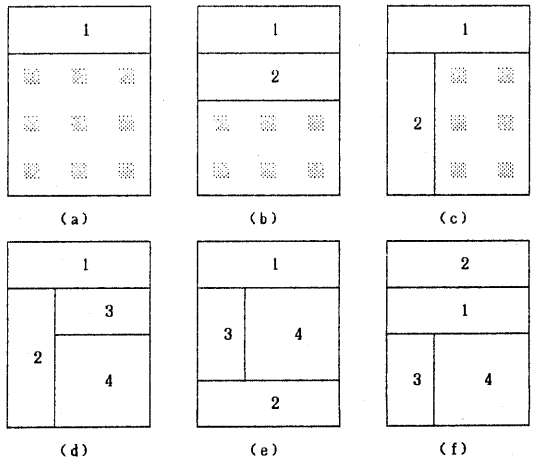


図11　配置状態の例（モデル３）

５．問題の Prolog 式記述

前述の図10に示す２次元ユニットと配置空間に対し、式(1) に示すルールを満たす解を探索する問題は例えば次のような Prolog 的表現が可能である。

```
frame = ((0,0,0),(0,0,0),(0,0,0),(0,0,0)).
piece1 = ((1,1,1)).
piece2 = ((1,1)).
piece3 = ((1,1),(1,1)).
SHAPE(unit1,piece1).
SHAPE(unit2,piece1).
SHAPE(unit3,piece2).
SHAPE(unit4,piece3).
rule_a:-NEIGHBOR(unit1,unit2).
rule_b:-NEIGHBOR(unit1,unit3).
rule_c:-NEIGHBOR(unit1,unit4).
rule_d:-NEIGHBOR(unit3,unit4).
rule_e:-NOT(rule_a).
rule_f:-NOT(rule_d).
rule:-(rule_b;rule_c),(rule_a,rule_f;rule_e,rule_d).
goal:-LAYOUT((unit1,unit2,unit3,unit4),frame,rule).
```

現時点ではこのような Prolog 的記述のインタプリタは未だ実現しておらず、直接的なデータを計算機が利用しやすい形に変換した後、これを用いて配置を実行している。プログラムは約1500の FORTRANステートメントより成り、パーソナル・コンピュータＰＣ９８０１Ｅ上で動作させている。

６．応用例

本手法を小住戸の間取り設計に応用した例を示す。図13の配置空間内に図12の室ユニットに対し下記のルールを適宜選択追加適用して住宅間取りを行なう場合に得られる解の総数と計算所要時間を表１に示す。

ルール １：玄関は北に１単位長以上面する。
ルール ２：居間は南に３単位長以上面する。
ルール ３：洗面所と浴室は１単位長以上隣接する。
ルール ４：洗面所と浴室は２単位長以上隣接する。
ルール ５：和室と押入れは１単位長以上隣接する。
ルール ６：和室と押入れは２単位長以上隣接する。
ルール ７：食堂と台所は２単位長以上隣接する。
ルール ８：玄関は食堂または居間と１単位長以上隣接する。
ルール ９：台所、洗面所、浴室、トイレは東または西または北に１単位長以上面する。
ルール10：台所、洗面所、浴室、トイレ、押入れは南に２単位長以上面さない。
ルール11：洗面所、トイレは居間または食堂と１単位長以上隣接する。

CASE 6で得られた現実的な解の一例を図14に示す。CASE 5より解総数が少ないCASE 6の方が計算所要時間が長いのはルール判定処理時間が配置探索の枝を刈ることにより短縮された時間を上まわったためと考えられる。

7．むすび

与えられた配置空間内に与えられた２次元図形ユニットを所定のルールを満たすように隙間なく配置する問題を解く手法について述べた。Prolog的に定式化した問題を解くインタプリタについては現在作成中であり別途御報告する予定である。

(1) 古川、近藤：”Two Dimensional Programming in Prolog”，情処学会記号処理研究会資料，83-23,1983.

(2) 加藤、村上、藤井：”住宅間取り設計エキスパート・システム”，情処学会知識工学と人工知能研究会資料，37- 1（昭和59年11月）.

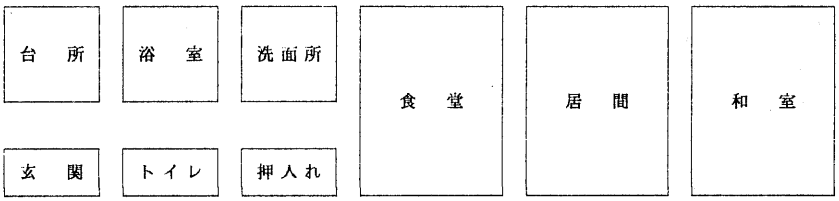


図12　室ユニット

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48
49	50	51	52	53	54

図13　室ユニットの配置空間

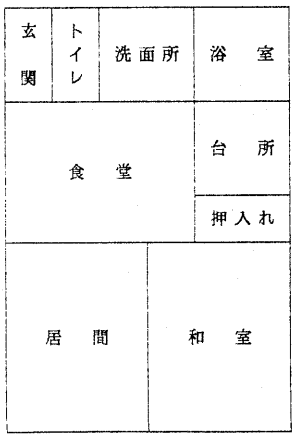


図14　解の例　(CASE 6)

表1　ルールの強化による解総数とCPU timeの変化

ルール No	1	2	3	4	5	6	7	8	9	10	11	解総数	CPU time
CASE 1												122688	1時間02分21秒
CASE 2	◎											25920	0時間14分03秒
CASE 3	◎	◎	◎		◎							1508	0時間04分52秒
CASE 4	◎	◎		◎		◎						420	0時間02分34秒
CASE 5	◎	◎		◎		◎	◎	◎				72	0時間01分13秒
CASE 6	◎	◎		◎		◎	◎	◎	◎	◎	◎	16	0時間01分20秒