Hyper Robot の解法プログラムについて

・ロボット 4 台の場合：HR4　(2010 年の GPCC の課題の一つ)

・ロボット 5 台の場合：HR5　(未稿)

2011.10.22　deepgreen

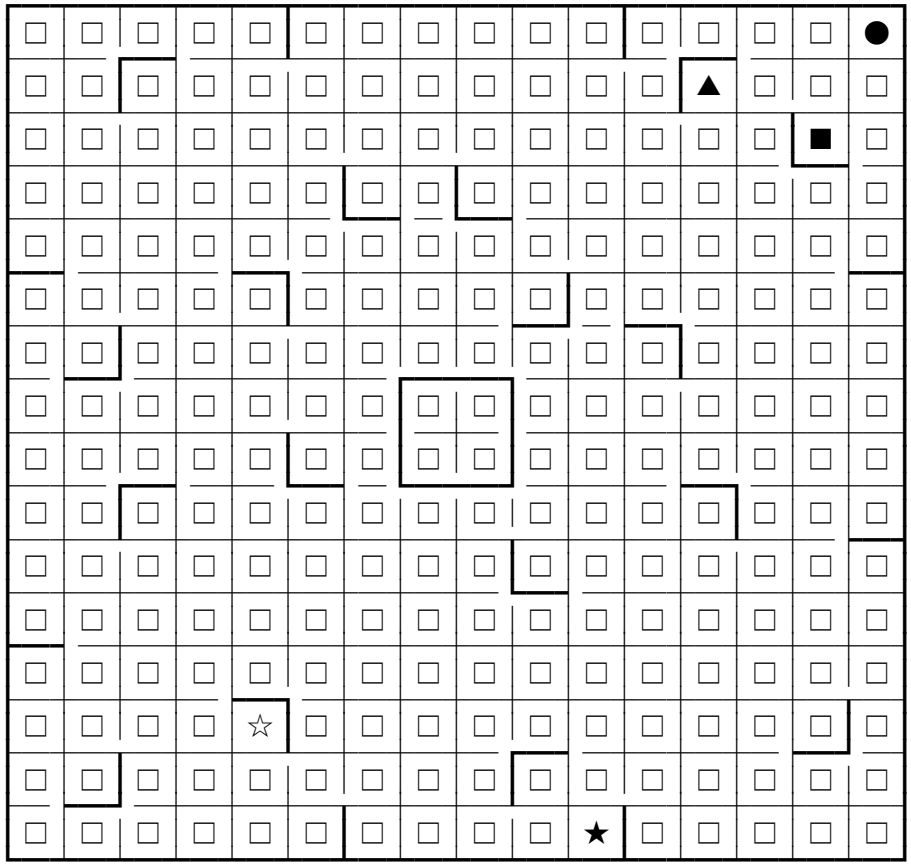
第Ⅰ部 HR4

・Solver():4 台のロボットの位置が指定されたとき、ゴールへの最短手数を求める探索

・MaxPath()：ゴールの位置が指定されたとき、ゴールへの最短手数が最長となるロボットの配置を求める探索

・MaxGoal()：MaxPath でゴールの位置を変えて、最長手数が最大となるゴールを求める探索

０．GPCC２０１０の問題



☆：Goal
★：Goal を目指すロボット
▲：
■：　一般のロボット
●：

1）上記のロボットの配置の場合、ゴールに最短手数の手順を求めよ　→ Solver

2）ロボットの配置を変更して、最短手数が最大となる配置を求めよ　→ MaxPath

１．準備

・用語の説明

ボード上のロボットの位置：０～２５５(縦方向を０～１５、横方向を０～１５)

R0：ゴールを目指すロボットの位置。

R1,R2,R3:一般のロボットの位置。ただし、ロボットの種別は区別しない。すなわち、(R1,R2,R3）の位置のロボットが、(▲,■,●）と（●▲,■,）とは同じ扱いとする。(働きが同じであるから)

C（n、m）：n 個の中から m 個を取り出す組み合わせの数。ただし、n＜mのとき、C(n,m)=0 とする。

・局面の管理

4 台のロボットによって生成される局面の数は、概略２５６＊C（２５６，３）≒７０８M 個。deepgreen の PC は、32 ビットマシンなので、作業域は１．５GB 位が限界となっている。従って、1 局面を１バイトで管理すれば、すべての局面をメモリに常駐させることができる。しかし、１局面を３バイト以上で管理する場合は、すべての局面のメモリ常駐は不可能となる。

・局面のハッシュ値：４台のロボットの位置からハッシュ値を求める関数。ロボットの位置とハッシュ値とは、１対１の対応をつける。

局面のハッシュ値 H = toPos(R1,R2,R3)*256 + R0

H_k：初期局面からｋ手進んだ局面のハッシュ値

H_0：初期局面（初期のロボット配置の局面のハッシュ値）

H_G：ゴールに到達した局面のハッシュ値

{　H_k　}：初期局面からｋ手進んだ局面のハッシュ値の集合

・toPos 関数：R1,R2,R3 の位置を０～C（２５６，３）－１のハッシュ数に変換する関数。具体的には組み合わせ型完全ハッシュ関数になる。この関数とその逆関数の仕組みの説明は、M.Hiroi 氏の HP の memorandum を参照されたい。ただし、n が非常に大きいので、そのままでは効率がわるいので、以下のように変更している。

h（ハッシュ値）＝C(R1,３)＋C(R2,2)+C(R3,1)

ただし、R1>R2>R3 とする

また、n＜ｍのとき C(n,m)＝０に留意されたい。（用語の説明でそのように拡張している）

- toNum 関数：toPos 関数の逆関数。０～C（２５６，３）－１のハッシュ値からロボットの位置 R1,R2,R3 を求める関数。

n が大きいので逆関数も手間がかかる。C（２５６，３）≒２．７M 個とメモリも大量には必要としないので、配列 HT で覚えておくことにする。

HT[0] = 0x000102　　HT[1] =0x000103　　・・・

２．Solver

・幅優先探索を採用

最短手数を求める問題では、反復深化探索がよく使われる探索法であるが、HR 問題ではよい下限評価値関数(LBE:lower bound estimater)が作れなかったこと、および、次の MaxPath 探索で使用する関数を準備する目的で、幅優先探索を採用した。（幅優先探索の技法については割愛する。例えば、高橋謙一郎氏の HP（）を参照されたい。）

幅優先探索で留意すべき事項は１つの局面を何バイトで管理するかということ点である。４バイトを用意することができれば、１つ前の局面(のハッシュ値)を保持できるので、探索完了後の手順の表示が非常に簡単になる。しかし、今回の問題では、準備の局面の管理の項で触れたように、１局面に４バイトを確保できないので、１バイトで管理することにした。１バイトの情報には、その局面の最短手数を格納する。

｛　H_0 ｝→ ｛　H_1 ｝ → ｛　H_2 ｝ → ・・・　｛　H_G ：R0 がゴールに到達した局面をふくむ集合　｝

３．MaxPath

・幅優先探索を採用

　この種の問題に対応できる探索法としては、反復深化探索は使えず、幅優先探索が一般的である。また、すべての局面を保持しなければならないため、これをメモリ空間に常駐できる範囲が通常の適応範囲となる。

４．MaxGoal

この問題の興味は、

①最長手数はどこまでいけるのか?

　➔ 31 手が最大

②Goal の位置とロボットの配置によっては、解なし(どうやってもゴールへたどりつけない配置)が存在するのではないか？

　→ GPCC2010 のボードの構成では、どのような配置であっても解は存在する。

Goal の位置を変えながら、MaxPath 探索を行い、最長手数が最大となる Goal の位置を探すだけ。時間はかかるが、難しさはない。