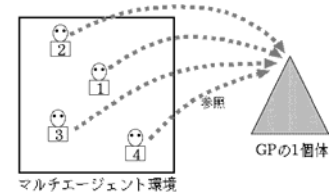
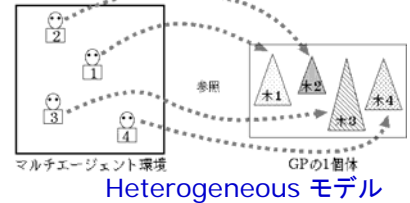


# 自律エージェントのグループ行動自動生成法 ADG

- 蟻などの社会性昆虫は、役割分担を決めて全体として極めて効果的な活動をしている。このような協調行動をマルチエージェントに行なわせる従来法として、遺伝的プログラミング(GP)を用いる方法がある。

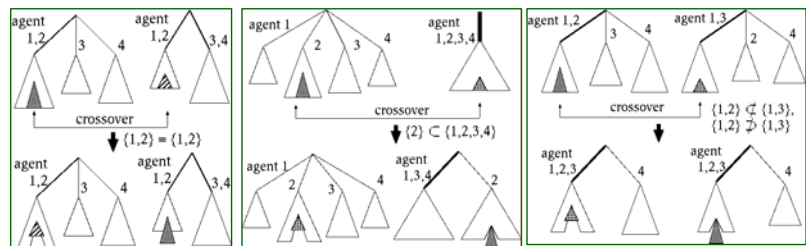


- GPを用いる場合、全エージェントを1つの木で制御するモデルと、エージェント毎に1つの木を用いるモデルがあるが、前者は複雑な行動を実現できず、後者では明示的なグループ分けが困難である。



- 提案手法 **ADG** (Automatically Defined Groups)

- グループ数, グループへの割り振り, 各グループの行動制御木の全てを進化過程で自動生成.
- エージェント集団をGPの1個体とする.
- 世代交代中にグループ数が(不変・増加・減少)する交叉を実行する.
- 最初はランダムなグループ数の個体集団から始め, 進化させる.

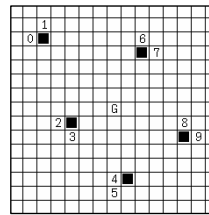


グループ数不変    グループ数増加    グループ数減少  
交叉におけるグループ数の増減の例

# エージェントの協調行動の獲得実験

## 問題設定:

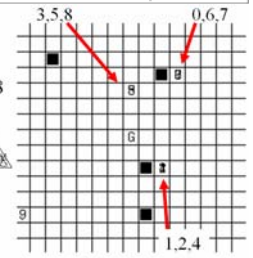
- 右のグリッド空間中にエージェント0~9を配置し, ■の荷物を押してゴールGに運ぶ。ただし荷物は重いため3人同時に押さなければ動かすことはできない。GPの木のノードの設定は右表の通り。



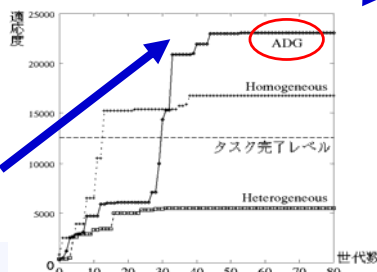
記号	機能
$+$ ( $x_1, x_2$ )	return ( $x_1 + x_2$ )
$-$ ( $x_1, x_2$ )	return ( $x_1 - x_2$ )
$\times 2$ ( $x_1$ )	return ( $x_1 \times 2$ )
$/2$ ( $x_1$ )	return ( $x_1 / 2$ )
$\rightarrow 90$ ( $x_1$ )	$x_1$ を時計回りに90度回転させる。
inv ( $x_1$ )	return ( $-x_1$ )
ifdot ( $x_1, \dots, x_4$ )	if ( $x_1 \cdot x_2 \geq 0$ ) return $x_3$ else return $x_4$
if $\geq$ ( $x_1, \dots, x_4$ )	if ( $ x_1  \geq  x_2 $ ) return $x_3$ else return $x_4$
Last	自分に対するGP木の最後の出力ベクトル。
Goal	自分からゴールへのベクトル。
Nearest-Tile	自分から最も近いタイルへのベクトル。
Same-Group	同じグループに属するエージェントのうち、自分から最も近いエージェント (ただし同じ位置にいるものは除く)へのベクトル。

## 実験結果:

- 獲得された木構造と, それによる行動の様子を右に示す。3人のエージェントが集合し, 協力して荷物をゴールに運ぶ協調行動を全自動で獲得した。
- エージェント3,5,8が参照するプログラムを示す。非常に複雑で人間が作ることは困難であることがわかる。



- 従来法よりも完成度が高い協調を実現することができた。



```

エージェント 3, 5, 8 が参照するプログラム
(+ (->90 (+ Same-Group Nearest-Tile)) (ifdot Nearest-Tile (- (if=> Nearest-Tile (if=> (+ (+ (ifdot Nearest-Tile (->90 (inv (+ (->90 Same-Group (inv (ifdot (if=> (/2 Nearest-Tile) Goal) (->90 (*2 (+ Same-Group Nearest-Tile))) (ifdot (inv Nearest-Tile) (if=> Goal (*2 Nearest-Tile) (inv (/2 Goal) (ifdot (->90 Same-Group) (if=> Last Goal Last Same-Group) (->90 Last) Last) (- (*2 (if=> Same-Group Last Nearest-Tile Same-Group) (->90 (+ Goal Last) (ifdot (if=> (if=> Goal Same-Group Goal Nearest-Tile) (- Goal Nearest-Tile) Last Goal) (- (+ Goal) Nearest-Tile) (- (/2 Same-Group) (inv Last) (inv (/2 Goal) (ifdot Goal Last Last) (ifdot Same-Group (->90 (- Same-Group (*2 Last) (->90 (+ Goal Last) (ifdot (if=> Nearest-Tile Same-Group Last Last) (ifdot Goal Last Same-Group Last) Same-Group) (ifdot (+ (/2 Last) (- Nearest-Tile Same-Group) (inv Same-Group) (*2 Last) Nearest-Tile Last) Goal) Last) (ifdot (inv (+ (->90 (->90 Last) (ifdot Same-Group Nearest-Tile Last (inv Same-Group) Last) Last) (if=> (ifdot (ifdot (if=> Nearest-Tile Last Same-Group Last Last) (ifdot (if=> (if=> Nearest-Tile Goal Last Last) Nearest-Tile) Same-Group) (inv Goal) Last) (->90 Last) Nearest-Tile) Same-Group Nearest-Tile Nearest-Tile) (- Nearest-Tile Same-Group) Nearest-Tile)))
    
```