Laboratoire d'Informatique de Paris 6
8, rue du Capitaine Scott
75 015 Paris France
el.Landau,Olivier.Sigaud,Pierre.Gerard}@l
http://miriad.lip6.fr/~landau
http://animatlab.lip6.fr/~{sigaud,pgerard
Laboratoire d'Informatique Fondamentale de L
Cité Scientifique
59 655 Villeneuve d'Ascq Cedex, France
Sebastien.Picault@lifl.fr
http://www.lifl.fr/~picault

In this paper we present ATNoSFERES, a nev
n indirect encoding Genetic Algorithm which l
nata controllers able to deal with perceptual ali
our ongoing line of research, we compare it wi
sed extension of the most studied Learning Class
gh two benchmark experiments. We focus in p
te generalization, and add special purpose feat
to fulfill that comparison. We then discuss the
state generalization in the experiments studied.

utionary Algorithms, Learning Classifier Sy
state generalization, ATN[1]

ion

lassifier Systems (LCS) [5] are used to tackl
ptive agents are involved in a sensori-moto
h agents perceive situations through their se
, each representing a perceived feature of the
s is to *learn* the optimal policy – *i.e.* which
, in order to fulfill their goals the best way t
ment Learning (RL) framework [20], the go
nize the scalar rewards it receives from its
oy a set of rules – or classifiers – specifying th
e *conditions* concerning the perceived situati

"Augmented Transition Networks"

ramework, explicit internal states were adde
) pair of the classifiers, e.g. in XCSM [14, 2
e additional information required to choos
on-Markov. The problem of properly settin
to *Genetic Algorithms* (GA).
we extend our first comparison presented in [1
ES". The latter is a new system that also
the behavior of agents facing problems in w
ors of attributes, and have to select actions
how in [12] that such an evolutionary approa
environments; in ATNoSFERES, the goals
(instead of classical LCS learning technique
ction, we present the features and properti
relies upon oriented, labeled graphs (§ 2.1)
action selection procedure. The specificity of
raph from a bitstring (§ 2.2) that can be ha
g of a GA, with additional operators. Then
esentation is formally very close to LCS rep
XCSM (§ 3.1). We remind the results of ou
in [12] (§4), and the comparison we made (§
ack of internal state generalization in ATNoS
y of the solutions that were found. A comp
oSFERES with and without a way to repre
6) allows us to discuss in detail the validity
usion, we present further additions that cou
ach an even higher performance (§8).

on of the ATNoSFERES model

sed expression of behaviors

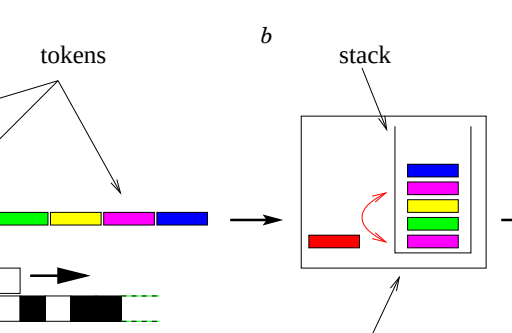
provided by the ATNoSFERES model [11, 18
is basically an oriented, labeled graph with
(or final) node (see figure 5). Nodes represen
ons of an automaton.
TNoSFERES binds conditions expressed as
endowed with the ability to generalize cond
But in ATNoSFERES , the conditions and a

step, the agent crosses an edge:
es the set of eligible edges among those star
. An edge is eligible when either it has no co
tions on its label are simultaneously true.
s randomly chosen in this set. If the set is emp
randomly over all possible actions, the cur
d, and we do not perform the next two steps
ns on the label of the current edge are seque
stem. Assuming that only one action can be p
the last action is actually performed. When
is empty, an action is chosen randomly.
current node becomes the destination of the
ops when it is at the *End* node (E). This
model and may never be reached. This appe
owing experiments (since agents reaching th
hus have a very low fitness, see § 4).

bed how the graphs are used, we now pres

n-building process

bing the behaviors is built from a genotype
asic structure containing only the *Start* and
ny different evolutionary techniques to aut
s circuits [8], finite-state machines [2], neura
Very roughly, we can sketch an opposition b
s that use the genotype as an encoding of a
orithms [5, 1, 3] or Evolutionary Strategies
oaches that use a single structure both as the
as Genetic Programming [7, 17], Evolution
], developmental program trees, e.g. [6, 4, 16
FERES model [10], we try to conciliate adv
es: on the one hand, since the behavioral phen
ion of a graph, we want it to be of any compl
ne-grain genotype (a bitstring) to produce it
tion of the solution space through "blind" ge



of the genetic expression we use to produce th
genotype. The string is first decoded into tok
cond step as instructions (b) to create nodes, ed

anslation is a simple process that reads the l
to a sequence of *tokens* (symbols). It uses t
on $\mathcal{G} : \{0,1\}^n \longrightarrow \mathcal{T}$ $(|\mathcal{T}| \leq 2^n)$ where $\mathcal{T}$ is
rent roles of which will be described in the
e number of available tokens, the genetic co
. Binary substrings of size n (decoded into

Tokens are instructions of the ATNoSFERI
le 1). They are interpreted one by one, while
n stream produced by the translator. The inte
perates on a stack in which parts of the futur
of the graph takes place during this interpre
nd connections, and connecting them to the
other stack-based languages (*e.g.* Forth, Pos
also be directly accessed by some instructi
by other means that only push/pop operati
pe with a "blind" evolutionary process (i.e.
ine-grain genotype), the graph built by th

uence of tokens is meaningful, the graph-bu
any variations affecting the genotype, thus t
mantical constraint on the genetic operators
is is to some extent order-independent and
very different genotypes.

resulting actions

s	manipulate the stack
o	no action, the token is just discarded
o	swap the two first tokens
o	push a copy of the first action or condition tok
l	delete the first action or condition token
e	push a copy of the first node token
e	delete the first node token[a]
l	pop the token, and puts it on the bottom of th
l	take the token from the bottom of the stack, a
s	**create nodes and connect them with edges**
e	create a new node and push it
t	create an edge from the first to the second noc label the edge with the set of conditions token of actions token until the second node, delete the action and condition tokens that we
t	create an edge from the Start node to the first label the edge with the set of conditions token of actions token until the node, delete the action and condition tokens that we
t	create an edge from the first node token to the label the edge with the set of conditions token of actions token until the node, delete the action and condition tokens that we
s	**actions and conditions tokens, specific to the a**
?	push the condition on the stack push the action on the stack

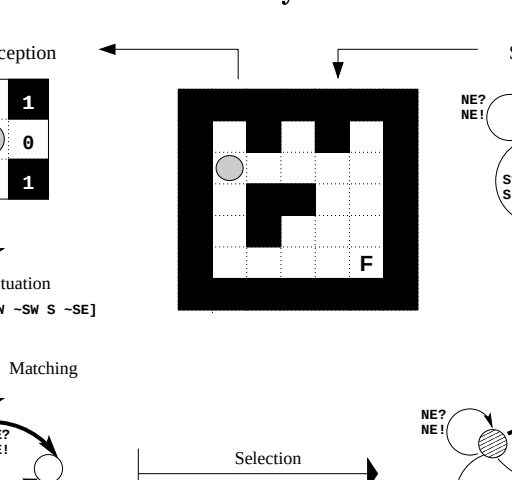
h building language. Here "first" (node, action or
(node, action or condition) token encountered v

r: possible other copies of the node still remain i
be copies of the same node, so it would be a self

nt abilities. Some of these tokens use toke

(actions, conditions), that are specific to an a
These token are just pushed onto the stack.

n into an evolutionary framework

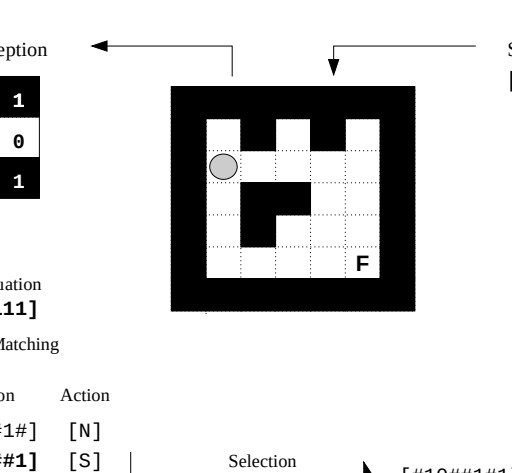


mple, the agent, located in a cell of the maze,
ocks in each of the eight surrounding cells. It ha
adjacent cells it should move. From its current
N ¬NW ¬W ¬SW S ¬SE] (token E is true when the
state (node) of its graph, two edges (in bold) ar
heir label match the perceptions. One is random
east) is performed and the current state is upda

the ATNoSFERES model has been applie
to produce controllers for agents.
ch agent has a bitstring genotype from which
c code depends on the perception abilities of
perform). The fitness of each agent is compu

f may change. Nodes or edges can in fact be
ry process, as can condition/action labels.

Classifier Systems



perceives the presence/absence (resp. 1/0) of b
cells (considered clockwise, starting with the no
, the agent perceives `[01010111]`. Within the lis
CS first selects those matching the current situat
g classifiers and the corresponding action is per

in the introduction, the problems tackled b
t that situations are defined by several attri
erties of the environment. A LCS has to buil
or of the system as shown in figure 3. Withi
"#" symbols in the condition parts of the cl
nce *don't care* symbols make it possible to us
several situations. Indeed, a *don't care* sy
f the considered attribute.

FERES model, a Pittsburg style LCS evolv

ry, in the *Michigan* style, the GA evolves a p
list of classifiers of a single agent. Here, thi
ed and modified. A fitness is associated to
e kept. Thus Michigan style LCS use GA
sifiers are improved during the life time of t
utility functions that depend on scalar rew
defined in the RL framework [20].
early LCS [5], the fitness was defined directl
to the classifier. After having defined a very
on found much more efficient to define the fi
he utility prediction. The resulting system,
used LCS to solve Markov problems.

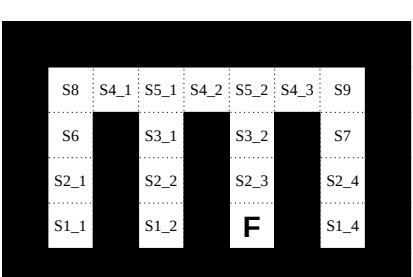
ple `Condition-Action` classifiers does not en
ave optimally in perceptually aliazed proble
happen that the current perception does n
ways choose the optimal action: as soon as t
n in different states, it will choose the same
appropriate in some of these states (see figu
blems, it is necessary to introduce internal s
way to probalistically link classifiers in orde
trast, Lanzi [14] proposed XCSM, where M s
f XCS with explicit internal states. XCSM m
composed of several bits that explicitly rep
. The memory register provides XCSM wit
l perceptions. Thus, dealing with perceptua
g information from the past experience of the
classifier contains four parts (see table 6): an
on, an internal condition about the internal
n in the environment and an internal action
.
condition and the internal action contain a
in the memory register. In order to be sele
match with both the external and internal
e LCS performs the corresponding action in

al states in XCSM and make ATNoSFERES
ng. Thus it is natural to compare ATNoSFE
ATN are characterized by several informatic
classifiers: the source and destination nod
valent to the internal condition and the in
ated to the edges correspond to the external
ions associated to the edges correspond to th

eriments

eptual aliazing problem

pose was to compare the evolutionary use
respect to their ability to deal with non-Ma
that comparison, we experimented our mo
which [14] provides empirical results obtaine



10 environment. **F** represents the goal to reach
of the maze; a few cells are unambiguous (S_i) b
al situations may require either similar actions o
$_{,4\}}$ but go south in S_{2_3})

ntal setup

oduce an experimental setup close to that us
ent, with regards to the specificities of our n

ent involves the following steps:

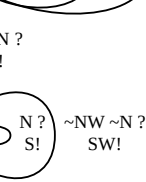
population with $N = 300$ agents with rando
eration, build the graph of each agent and

lividuals with higher fitness (namely, 20 % c
new ones by crossing over the parents. The
mutations (with a 1% rate) and insertions or c
rate) on the bitstring of the offspring.
rocess with the new generation.

aluate the individuals, they are put into the e
ll in the grid, and they have to find the foo
The agent cannot perceive the food, and it
ne step; when this action is incompatible wit
n the west cell contains an obstacle), it is sim
me step and stays on the same cell). Its fitne
$+ 2 * R$ (F: fitness for the run; D: number c
ered during the run; K: time steps spent on al
ne food is found; R: remaining time if the fo
imit). Thus, the selection pressure encoura
tion. At each time step, the current cell is a
lls (in order do compute D and K). The term
ıding an already-known cell is still preferred
cells. Each agent is evaluated 4 times starti
l fitness is the sum of the fitnesses compute
with $B = 30$ and a 20 time steps limit, the
nts reported here were carried out on variou
540 bits. The original population genotype s
xperiment has been bounded by 10,000 ger
cases to reach high enough fitness values.

a behavioral graph obtained by the best in
iment. It has also been represented in a L

ose graph is described in figure 5 has the f
corridor, it first reaches the horizontal corr



#	0	#	#	#	#	#
0	#	#	#	#	#	#
1	#	1	#	#	#	#
#	0	#	#	#	#	#
#	#	1	1	#	#	#

Fig. 6. A LCS-like rep
graph on figure 5. E
tions, IC: internal con
nal actions, IA: interna

the best individual in a
eriment

y goes straight to the food. This is a nearly
are clear distinctions between the bottom of
cells $S_{\{1,2\}\_n}$), the top of vertical corridors (N
orridor (E $\rightarrow$ $S_8, S_{\{4,5\}\_n}$) and the crucial
¬N ¬NW).

n of the first experiments

nted a discussion resulting from the compa
I. The main points we made were the follow

epresentation: While XCSM produces a co
ich the size of the external conditions part a
hosen in advance, ATNoSFERES builds a gra
and labels on the edges can be minimal t
that we focus on the best agent only). Hen
S can be minimal while XCSM model cannot

earning and Classifier Selection: One impor
t to ATNoSFERES is that the forces of cla
gorithm. In order to remedy the fact that A
necessary to include into the fitness funct
nation about the actual behavior of the age

will be discussed next: ATNoSFERES canno
on rules that can be fired whatever the intern
M with an internal condition composed of "#

n XCSM, a # in the internal condition allows
the internal state represented by the memor
its action regardless of the internal state. Or
been chosen in those first experiments (see ta
dealing with a default behavior, since connec
between two nodes (i.e. two internal states). V
6.

lts given in [12] showed that the behavior ob
ES was not completely optimal, and that obt
ire a major structural change in the graph w
antage.

eriments

g the need for state generalization

ded from the previous discussion by assuming
to deal with a default behavior, regardless of
tage of XCSM over our model. Therefore we
dd an internal state generalization property
ph building language, with a new *defaultSelf(*

n resulting actions

s create nodes and connect them with edges

: creates an edge from all the already present no
labels the edges with the set of conditions toke
of actions token until the first node,
deletes the action and condition tokens that w

on to the graph building language (see table 1
node token encountered while going down the sta

that, since only the already created nodes wil
generalization depends on the time when t
xample, if a *defaultSelfConnect* comes befo
h are created, then the next nodes will not
struction.

figure 7, self-connecting transitions on node
in internal condition and/or internal actio
s the figure shows, if only the transition in n
o a completely specified internal condition,
ent in all nodes, it is equivalent to a comp
and internal action.

estick experimental setup

experiments on Maze10, the best fitness v
eoretic fitness. Since these results are very c
nts do not provide a large enough opportunit
he efficiency of the new encoding. Therefore,
naze (see figure 8) where the advantage of t
uage should be more significant with respect

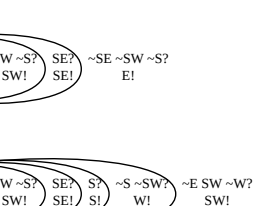
rts from the top cells of any of the vertical
ng locations only, because we are focusing on
han searching for a general behavior to solv
While going south along the "candles" from
pty cell on the side, the agent can determi
s once it reaches the bottom of the candles.
-right and far-left candles.
tate management strategy we have envisione
following. The agent just needs one bit of
gent sees an empty space on its left hand s
epresents whether it is in the left part or
This informations suffices to choose the rig
ottom of the candles. Given this internal s
essity to generalize on the internal state val
rs all the $\{S_{a\_i}\}$ cells. In all those cells, whi
eptual conditions, the agent must go south, *i.*
t is in the left part or the right part of the

ndlestick environment. **F** represents the goal to
res; 5 cells are unambiguous (S_i). In the other
may require either similar actions or different o
o east in $S_{g\_\{6,7,8,9,a\}}$)

NoSFERES, this means that it must follow tl
rnal state is.
f this property of the 12-Candlestick maze
rnal state management strategy presented ab
nce. Any other strategy implies that the agen
hoose the correct directions. Indeed, disambi
right corners like the strategy obtained in M
r with respect to the optimal path. But, as it
of that paper, even if this optimal behavior
using the *defaultSelfConnect* token, it is nev
t.
ntal setup is similar to the one in our previo
r the genetic encoding and the fact that tl
e empty cells. Like before, we need at least
me in order to observe the influence of our
ncodings. In the first encoding, we use the *n*
replaced by the *defaultSelfConnect* token.
evaluated 6 times starting in each of the 1
ion is the one of Maze10 experiments. In
a 30 time steps limit, the fitness is 6732.
made the experiments for 5 different initial
0 and 540 bits). During the experiments, the
lengths of the genotypes.

estick experiment results

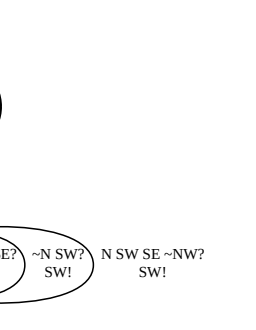
respectively show the best solutions found
nnect token. The fitness of both these indivi
2. The missing points are lost when the age
foot of the far-right candle (see figure 8, c



tick. Graph of the best
e *defaultSelfConnect* to-
pears 3 times, after the 2
, so there are 3 identical
ges for both nodes.

0	#	#	1	#	#	1
#	#	#	#	#	#	1
#	#	#	#	#	0	0
0	#	#	#	0	1	#
#	#	1	0	#	#	1
#	#	#	#	#	1	0
#	#	#	#	#	#	#

Fig. 10. A LCS-like re
graph on figure 9. The t
correspond to self-conn
graph. EC, IC, EA, IA



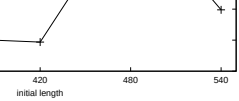
estick. Graph of the best
g the *defaultSelfConnect*

EC					
E	NE	N	NW	W	SW
#	#	#	#	#	#
#	#	#	#	#	0
#	#	#	#	#	#
#	#	0	#	#	1
#	#	1	0	#	1
#	#	#	#	#	0
0	0	#	#	0	#
#	#	#	#	#	0
#	#	1	#	#	#
#	#	#	#	#	#

Fig. 12. A LCS-like r
graph on figure 11. E
figure 6.

n of 12-Candlestick results

ws, the average fitness obtained with the
Candlestick experiment is significantly better
hat token. The difference between the perf



estick. Average fitness
self token

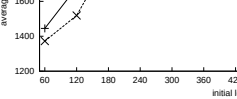
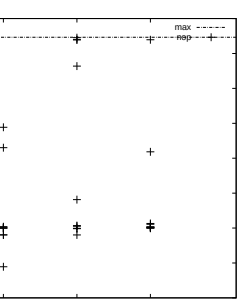


Fig. 14. Maze10. Avera
without self token

o the different initial lengths of the bitstrin
e can clearly see from these figures that the
elective advantage to our agents.



estick. Fitness without
token

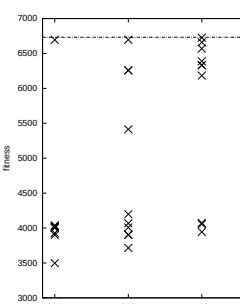


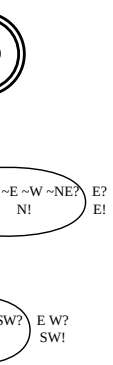
Fig. 16. 12-Candlestic
faultSelfConnect token

of making these experiments on the 12-Canc
faultSelfConnect token is working well and
was designed. But, since the 12-Candlestick
the use of the *defaultSelfConnect* token, it
or not this property can be generalized to M
cant results.

laze10 experimental setup

l setup is that of our previous Maze10 exp
for the genetic encoding. Like in the previou

res 17 and 19. The best fitness for both genet
defaultSelfConnect token) are close to each
r a maximum of 4500. This best fitness was
st, and only one time for the second, over t
tic encoding. As in our previous Maze10 exp
t when reaching the NE corner when comin
her points are also lost when the agent goes
ng through the cell on top of it, instead o



Graph of the best individ-
ultSelfConnect token for
n appears 1 time, and is
le only.

EC						
E	NE	N	NW	W	SW	S
1	#	0	#	#	#	#
0	#	#	#	#	0	#
#	1	#	#	#	#	#
0	0	#	#	0	0	#
1	#	#	#	#	#	#
0	#	0	#	#	#	#
0	#	#	#	#	#	#
#	#	0	#	#	0	#
1	#	#	#	1	#	#

Fig. 18. A LCS-like re
graph on figure 17. E
figure 6.

iscussion

nd 14, it is clear that ATNoSFERES obtains b
lestick and on Maze10 with the *defaultSelfC*

behavior is never reached, however (the bes
um). The best individual on the 12-Candles
e *defaultSelfConnect* token as expected, so
ecified internal state. This result seems to s
ding to which being able to generalize on th
ve such behavioral problems, and results or
rality of the assumption.

#	0	#	1	#	#	#
0	#	0	#	#	#	#
1	#	0	#	#	#	#
#	#	#	#	#	#	#
#	#	1	#	#	1	#

Fig. 20. A LCS-like re
graph on figure 19.EC
figure 6.

Graph of the best individ-
efaultSelfConnect token.
010 and 100 is actually

xamination of the best individuals obtained
that our assumption must be refined. In p
aze10, presented in figure 17, uses the *defa*
s only one node in the stack. More generall
nnect token has been used many times to
tion in the graph, or none at all (if no nod
ct is interpreted).

in figure 7, since the interpretation of that t
onnecting transitions only to the nodes alre
t always result in the equivalent of full gen
particular, when there is only one self-con
interpretation of the *defaultSelfConnect* to
lly specified internal condition part followed
art.

however ATNoSFERES can obtain one se
e *defaultSelfConnect* token. But doing so r
on the bitstring are fulfilled. Thus a self-
o happen without the *defaultSelfConnect* to

hat ATNoSFERES gets a better performanc
n than without it just because having self-
ial property and having that token significa
ve that property.

his more detailled study seems to reveal is t
e necessary to solve a non-Markov problem, it
he possibility to specify (condition, action) t
o change its internal state.

f a comparison between XCSM and ATNoS
aper the importance of the ability to represe
order to do so, we have introduced a new
a self-connecting transition to all the nodes
ve also presented a new maze experiment sp
ıs able to generalize on the internal state.
nts have shown that the performance of ou
th this addition. But, while this result seer
being able to generalize on the internal sta
tive algorithms, a closer examination of wha
ments reveals that our *defaultSelfConnect* t
nt purpose than just generalizing on the in
ave another interesting property than the on

able to conclude more accurately on the rela
generalization property and the stable inter
s will be necessary. We believe that going i
son between XCSM and ATNoSFERES on th
will help identifying further what is really
vior. In particular, we should try to assess th
on the internal condition and on the interna
o specialized tokens representing each prope

ONG, *An Analysis of the Behavior of a Class o*
thesis, Dept. of Computer and Communication
1975.
, A. J. OWENS, AND M. J. WALSH, *Artificial*
olution, John Wiley & Sons, 1966.
BERG, *Genetic Algorithms in Search, Optimiz*
dison-Wesley, 1989.
Neural Network Synthesis Using Cellular Encodi
n.D. thesis, ENS Lyon – Université Lyon I, 1994
ND, *Adaptation in Natural and Artificial Systen*
Applications to Biology, Control, and Artificial
an Press, Ann Arbor, MI, 1975.

. 151–170.
D. E. Goldberg, D. B. Fogel, and R. L
ming 1996: Proceedings of the First Annual C
A, 1996, MIT Press.
and S. Picault, *Stack-Based Gene Expression*
1, LIP6, Paris, 2002.
S. Picault, and A. Drogoul, *ATNoSFERES*
Behaviors, in Proceedings of the AISB'01 Symp
Iulti-Agent Systems, 2001.
S. Picault, O. Sigaud, and P. Gérard, *A*
S and XCSM, in Langdon et al. [13], pp. 926–93
n, E. Cantu-Paz, K. Mathias, R. Roy, D
hnan, V. Honavar, G. Rudolph, J. Wegene
Schultz, J. F. Miller, E. Burke, and
f the Genetic and Evolutionary Computation C
ork, july 9-13 2002, Morgan Kaufmann.
An Analysis of the Memory Mechanism of XC
Genetic Programming Conference, 1998.
ayer, *Mathematical Models for Cellular Interacti*
, Journal of theoretical biology, 18 (1968).
L. Spector, *Evolving Graphs and Networks w*
Report, in Koza [9], pp. 117–124.
na, *Strongly Typed Genetic Programming*, in Ev
1995.
nd S. Landau, *Ethogenetics and the Evolution*
Proceedings of the 5th World Multi-Conference
Informatics (SCI'01), N. Callaos, S. Esquivel,
pp. 528–533.
efel, *Evolution and Optimum Seeking*, John W
n and A. G. Barto, *Reinforcement Learning, a*
idge, MA, 1998.
on and L. Bull, *CXCS*, in Learning Classifier S
pplications, P. Lanzi, W. Stolzmann, and S. Wi
lberg, 2000, pp. 194–208.
n, *ZCS, a Zeroth level Classifier System*, Evolutio
1–18.
n, *Classifier Fitness Based on Accuracy*, Evolutio
149–175.
os, *Transition Networks Grammars for Natural*
ons of the Association for the Computational M
ving Artificial Neural Networks, Proceedings of t