

SEMEION
CENTRO RICERCHE DI SCIENZE DELLA
COMUNICAZIONE

MASSIMO BUSCEMA

Genetic Doping Algorithm

(*GenD*)

ANNO 2000

EDIZIONI SEMEION

Technical Paper n. 22

MASSIMO BUSCEMA

Genetic Doping Algorithm

(*GenD*)

2001

EDIZIONI SEMEION
Technical Paper n. 22

Tutti i diritti sono riservati. Ogni tipo di riproduzione, parziale o integrale, anche in fotocopia, è vietata.

Tutte le procedure e i modelli presenti nel paper sono originali del Semeion e non possono essere informatizzati senza previa autorizzazione scritta del Centro Ricerche Semeion, che possiede i programmi informatici protetti.

Copyright © SEMEION 2000

Via Sersale 117/119 – 00128 Roma, Italy

Sommario

<i>Capitolo primo:</i> Caratteristiche Fondamentali	pag. 5
<i>Capitolo secondo:</i> Criterio di vulnerabilità	» 5
<i>Capitolo terzo:</i> Criterio di accoppiabilità	» 6
<i>Capitolo quarto:</i> Criterio dell'ultima opportunità	» 6
<i>Capitolo quinto:</i> Il nuovo Cross Over	» 7
<i>Capitolo sesto:</i> Le caratteristiche di GenD	» 9
<i>Capitolo settimo:</i> Confronto tra GenD e Genetic Algorithm tradizionali (G.A.)	» 12
<i>Capitolo ottavo:</i> GenD & TSP	» 15
8.1 Il problema	» 15
8.2 La codifica del problema	» 16
8.3 Gli adattamenti di GenD	» 16
8.4 Gli esperimenti	» 18
8.5 I risultati	» 19
<i>Riferimenti bibliografici</i>	» 28

Genetic Doping Algorithm (*GenD*)

Massimo Buscema (2000)
Direttore del Centro Ricerche Semeion di Roma

1. Caratteristiche fondamentali

L'algoritmo *GenD* ha l'obiettivo di rendere più **naturale**, e meno centrato sul concetto culturale di liberismo individuale, il processo evolutivo di popolazioni artificiali; ciò allo scopo di migliorare la **velocità** e la **qualità** delle soluzioni che si tenta di ottimizzare.

L'algoritmo *GenD*, come primo passo, calcola il punteggio di fitness di ogni individuo, in accordo alla funzione che si intende ottimizzare. Il valore della **salute media** dell'intera popolazione non è, però, un semplice **indice**.

In *GenD* la salute media costituisce il criterio di **vulnerabilità**, prima, e di **accoppiabilità**, dopo, di tutti gli individui della popolazione ad ogni generazione e ad ogni generazione muta.

Quindi, *GenD* è un algoritmo che non ha nell'individuo la propria unità fondamentale, come gli Algoritmi Genetici e molti altri algoritmi evolutivi. Per *GenD*, l'unità di riferimento è la **specie**, che agisce, ad ogni generazione, sull'evoluzione degli individui sotto forma di salute media [Eldredge, 1995].

Il feedback loop tra individui e salute media della popolazione (specie) permette, in *GenD*, di trasformare evolutivamente la popolazione stessa da una **lista di individui** (come nei sistemi evolutivi tradizionali) in un **sistema dinamico di soggetti**.

2. Criterio di vulnerabilità

Tutti gli individui, la cui salute è **inferiore** o **uguale** alla salute media della popolazione, vengono iscritti in una **lista di vulnerabilità**. Non vengono, perciò, eliminati, ma continuano a partecipare al processo; tuttavia sono **segnati**.

Il numero degli **individui vulnerabili** stabilisce automaticamente **il numero massimo di matrimoni** permessi in quella generazione.

Il numero di matrimoni possibili ad ogni generazione, quindi, è variabile ed in funzione della salute media della popolazione.

Il terzo passo di *GenD* consiste nell'accoppiamento tra gli individui. Tutta la popolazione partecipa a questa possibilità. Il numero massimo di chiamate casuali

all'accoppiamento sarà uguale alla **metà** del numero di individui segnati come vulnerabili.

3. Criterio di accoppiabilità

Perché un accoppiamento generi prole, entrambi gli individui candidati devono avere una salute il cui valore sia **vicino** alla media di salute dell'intera popolazione (media $\pm k$, dove $2k$ definisce l'ampiezza della banda di accoppiabilità).

GenD può usare anche un altro criterio di accoppiabilità: ogni coppia di individui può generare prole se la fitness di almeno uno di loro è maggiore della media.

Gli effetti di questi due criteri sono molto simili: la distribuzione normale di ogni fitness individuale, rispetto alla media, lavora ad ogni generazione come una banda dinamica di cross-over.

L'algoritmo *GenD*, quindi, assume che gli individui “sui generis”, troppo deboli o troppo sani, tendano a **non sposarsi** tra loro. In pratica, il matrimonio non si addice ai **migliori** e ai **peggiori**. Sono i soggetti “più normali” che tendono a sposarsi.

Inoltre, in *GenD* esiste il principio del **Sex Free**: sono ammessi “incesti” e “poligamie”.

I “figli” di ogni matrimonio vanno ad occupare i posti dei soggetti segnati nella lista di vulnerabilità. Può accadere, quindi, che un individuo debole abbia la possibilità di continuare ad esistere tramite la propria prole.

4. Criterio dell'ultima opportunità

Il numero di matrimoni possibili è stabilito in funzione del numero di soggetti segnati come **vulnerabili**, e della **salute media** della popolazione. Il criterio di accoppiabilità, tuttavia, spinge il sistema ad avere un minore numero di matrimoni celebrati rispetto a quelli possibili. La differenza tra matrimoni disponibili e matrimoni realizzati definisce il numero delle **mutazioni**.

Vengono, così, mutati quei soggetti presenti nella lista di vulnerabilità che non sono stati sostituiti dai figli generati dai matrimoni realizzati precedentemente.

A questo numero variabile di soggetti deboli, quindi, viene concessa, tramite una mutazione, un'**ultima opportunità** per rientrare nel gioco evolutivo.

5. Il nuovo Cross Over

Il *criterio di accoppiabilità* permette matrimoni tra gli individui solo quando:

1. entrambi hanno una salute i cui valori rientrino in una banda specifica

- intorno alla salute media della popolazione;
2. almeno uno possiede una salute i cui valori siano maggiori della media dell'intera popolazione.

Ciò significa che

$$\begin{aligned} SE \quad F_i &= \text{Salute individuo } i\text{-esimo;} \\ \bar{F} &= \text{Salute media;} \\ \sigma &= \text{Varianza della salute della popolazione.} \end{aligned}$$

ALLORA F_i è un candidato al matrimonio se:

$$k = 1 - \sigma;$$

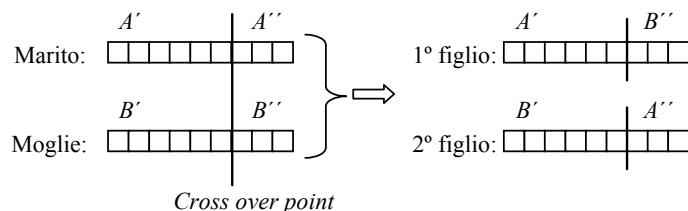
$$(\bar{F} - k) \leq F_i \leq (\bar{F} + k).$$

Nel secondo caso:

$$SE \quad (F_i \geq \bar{F} \cup F_j \geq \bar{F})$$

ALLORA i candidati possono generare prole.

L'accoppiamento, tuttavia, in *GenD* **non consiste** nel semplice scambio di geni tra marito e moglie intorno a un punto di incrocio (Cross Over); es.:



In *GenD* l'accoppiamento di geni tra genitori è effettuato in modo **selettivo**.

Abbiamo programmato due tipi di accoppiabilità:

1. una logica di Cross Over: quando sono ammesse ripetizioni;
2. un *opportunistic Cross Over*: quando non sono permesse ripetizioni.

La logica di Cross Over considera quattro casi:

1. *Salute del padre e della madre maggiori della salute media dell'intera popolazione*:

$$(P > m) \& (M > m)$$

2. *Salute di entrambi i genitori minore della salute media dell'intera popolazione*:

$$(P < m) \& (M < m)$$

- 3 e 4. *Salute di uno dei genitori minore e quella dell'altro maggiore della salute media dell'intera popolazione*:

$$(P > m) \& (M < m) \text{ oppure } (P < m) \& (M > m)$$

Nel **primo caso**, la generazione dei due figli (supponiamo, per semplicità, il caso con due figli e un solo punto di Cross Over) avviene nel modo tradizionale:

$$1. \quad (P > m) \& (M > m): \quad \begin{array}{c} A' + A'' \\ B' + B'' \\ \hline \text{genitori} \end{array} \Rightarrow \begin{array}{c} A' + B'' \\ B' + A'' \\ \hline \text{figli} \end{array}$$

Nel **secondo caso**, la generazione dei due figli avviene tramite la **negazione** dei geni dei genitori:

$$2. \quad (P < m) \& (M < m): \quad \begin{array}{c} A' + A'' \\ B' + B'' \\ \hline \text{genitori} \end{array} \Rightarrow \begin{array}{c} \overline{A'} + \overline{B''} \\ \overline{B'} + \overline{A''} \\ \hline \text{figli} \end{array}$$

Nel **terzo e nel quarto caso**, solo il genitore la cui salute è superiore alla salute media trasmette i propri geni, mentre quelli dell'altro vengono **negati**:

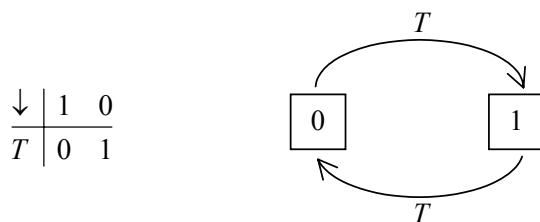
$$3. \quad (P > m) \& (M < m): \quad \begin{array}{c} A' + A'' \\ B' + B'' \\ \hline \text{genitori} \end{array} \Rightarrow \begin{array}{c} A' + \overline{B''} \\ \overline{B'} + A'' \\ \hline \text{figli} \end{array}$$

oppure:

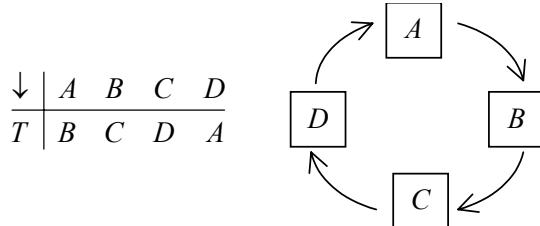
$$4. \quad (P < m) \& (M > m): \quad \begin{array}{c} A' + A'' \\ B' + B'' \\ \hline \text{genitori} \end{array} \Rightarrow \begin{array}{c} \overline{A'} + B'' \\ B' + \overline{A''} \\ \hline \text{figli} \end{array}$$

Il concetto di **negazione genetica**, in *GenD*, non corrisponde all'**annullamento** dei geni del genitore più debole, neppure ad un loro rimpiazzo effettuato casualmente. Si tratta, invece, di una sostituzione genica effettuata con il criterio di una **finestra mobile** che scorre, da destra o da sinistra, le **opzioni genetiche alternative** per ogni singolo gene in ogni specifica posizione.

Se, ad esempio, un certo gene presenta due sole alternative $g[0,1]$, allora la **finestra mobile** avrà la seguente forma:



Allo stesso modo si procederà con un gene che presenta quattro stati possibili; $g[A,B,C,D]$:



Il criterio della **negazione tramite finestra mobile** è applicabile anche quando i vari stati possibili di un gene non sono tra loro ordinati. Questo metodo, infatti, si fonda sulla sistematicità di esplorazione dello spazio delle fasi e sul mantenimento di uno stesso criterio di sistematicità.

L'opportunistic Cross Over lavora quando non sono permesse ripetizioni.

In questo caso, i genitori sono allineati rispetto ad un punto di Cross Over casuale; in seguito, è generata la prole selezionando il gene più efficace dei genitori. Questo algoritmo è ripetuto finché tutti i geni dei genitori sono completi; questa regola è molto simile ad una variante dell'algoritmo di Greedy.

In definitiva, il numero dei matrimoni e delle mutazioni in *GenD* non sono parametri esterni, ma variabili interne che si autodefiniscono in modo adattivo, tenendo conto della dinamica complessiva del “sistema popolazione”.

6. Le caratteristiche di *GenD*

La caratteristica più evidente di *GenD* consiste nel presentare un algoritmo per certi aspetti molto più **culturale**, e per altri più **naturale**, rispetto ai classici algoritmi genetici.

Ciò ci ha indotto ad immaginare una nuova famiglia di algoritmi adattivi ed evolutivi che abbiamo definito “**Algoritmi Naturali**”, in opposizione agli algoritmi “occidentali” presenti nella letteratura scientifica.

In *GenD*, il numero dei **matrimoni** e delle **mutazioni** varia di generazione in generazione. Questi valori sono tra loro connessi, ed entrambi dipendono dai valori della **salute media** della popolazione ad ogni generazione.

Queste caratteristiche rendono **più naturale** *GenD* rispetto alle soluzioni tradizionali: in *GenD* nulla è fisso, ma tutto dipende dal contesto e il contesto varia nel tempo.

In *GenD*, inoltre, l’evoluzione degli individui è **centrata intorno alla media** della loro salute; le soluzioni estreme, quindi, (i **migliori** o i **peggiori** individui) vengono considerate come **rumore** e non come criterio evolutivo.

Ciò non toglie che, dal punto di vista ingegneristico, gli individui migliori siano un **rumore utile**.

La promozione dei matrimoni tra individui medi, in *GenD*, ha degli effetti positivi:

- a. la salute media di tutta la popolazione cresce in modo lineare e compatto nel

tempo; questo significa che, per uno stesso problema, vengono fornite **più soluzioni ottime**.

In breve: la salute media tende alla salute del migliore individuo, per cui **molti individui** presentano soluzioni diverse ma tutte ottimali.

Inoltre, *GenD* ha un suo criterio di **convergenza**: quando la salute media coincide per più generazioni con la salute del migliore individuo, allora il **massimo** della funzione è stato raggiunto;

- b. *GenD* implica, senza bisogno di regole apposite, l'aumento della **biodiversità** degli individui, in modo proporzionale all'aumento della salute media della popolazione: più la salute media cresce, più aumentano gli individui vulnerabili, e, di conseguenza, i matrimoni e le mutazioni.

L'aumento della biodiversità, comunque, non incide sugli individui migliori (in genere non toccati dai matrimoni e dalle mutazioni), ma sulla media: questa, infatti, subisce una decrescita, per poi riprendere la sua ascesa.

In breve: all'aumento della biodiversità, le soluzioni migliori restano stabili, ma la riorganizzazione degli individui "medi" può produrre altre soluzioni, migliori o equivalenti a quelle precedenti.

Questo significa che *GenD* può uscire spontaneamente dai **massimi locali** che lo intrappolano, per mirare ai valori più prossimi al **massimo globale** della funzione.

In *GenD* la produzione spontanea di biodiversità ha un ulteriore vantaggio: non è necessario ricorrere alla **mutazione** per mantenere flessibili gli individui dell'intera popolazione.

Crediamo, infatti, che fondare la biodiversità, cioè la ricchezza della varianza, su un **errore casuale**, sia un errore sistematico. Nei processi naturali la diversità è **metodo, non contingenza**.

Il processo di mutazione, in *GenD*, viene collocato, invece, come un'**opportunità** per gli individui più deboli e non come motore evolutivo.

Le caratteristiche **culturali** di *GenD* sono altresì evidenti: i matrimoni non generano **figli tautologici** rispetto ai genitori. Le parti migliori dei genitori vengono trasmesse ai figli, mentre le caratteristiche più deboli vengono cambiate tramite una **esplorazione sistematica** dello spazio delle possibilità.

Questo mutamento sistematico ha due conseguenze importanti:

- a. trasforma i matrimoni in generatori di biodiversità;
- b. permette ai "nuovi nati" di **rubare** e fare proprie quelle caratteristiche genetiche, presenti nel contesto della popolazione, che si sono dimostrate efficaci.

Il rapporto dialettico, che *GenD* crea tra crescita della salute media e crescita della biodiversità, rende la dinamica del sistema **strutturalmente instabile**. Si tratta di una instabilità strutturale, ma tendenzialmente **locale**: solo gli individui la cui salute è uguale o inferiore alla salute media entrano nelle liste di vulnerabilità. E solo questi rischiano l'eliminazione.

Questa dinamica instabile obbliga il sistema ad un andamento ritmico, che per-

mette di “saltare”, da un massimo locale ad un altro, in modo spesso vantaggioso per l’aumento della fitness della funzione.

<i>G.A. classico</i>	<i>GenD</i>
<ul style="list-style-type: none"> • Valuta la salute individuale • Crea la ruota delle probabilità 	<ul style="list-style-type: none"> • Valuta la salute individuale
<ul style="list-style-type: none"> • Crea una nuova popolazione sul criterio della ruota delle probabilità 	<ul style="list-style-type: none"> • Crea una lista di vulnerabilità sul criterio della media
<ul style="list-style-type: none"> • Effettua una % fissa di matrimoni in funzione della ruota delle probabilità - (si sposano i migliori) 	<ul style="list-style-type: none"> • Effettua una % variabile di Matrimoni in funzione della lista di vulnerabilità (numero) e della media (qualità) - (si sposano i normali)
<ul style="list-style-type: none"> • Ogni matrimonio ha N Cross Over e incrocia i geni dei genitori (produce possibili miglioramenti) 	<ul style="list-style-type: none"> • Ogni matrimonio ha N Cross Over e consiste in una ricerca degli stati possibili tra genitori e popolazione (produce possibili miglioramenti e aumenta la biodiversità)
<ul style="list-style-type: none"> • La percentuale delle mutazioni è fissa e serve a produrre biodiversità 	<ul style="list-style-type: none"> • La % delle mutazioni è variabile, in funzione dei matrimoni non celebrati, e serve a offrire solo ad alcuni dei più vulnerabili un’ultima opportunità
<ul style="list-style-type: none"> • La biodiversità è fornita dagli errori (mutazioni) e decresce al crescere della media 	<ul style="list-style-type: none"> • La biodiversità è generata dai matrimoni e cresce al crescere della media
<ul style="list-style-type: none"> • Il sistema tende alla stabilità 	<ul style="list-style-type: none"> • Il sistema diventa instabile quanto più si approssima alla stabilità
<ul style="list-style-type: none"> • Non c’è evoluzione dell’evoluzione 	<ul style="list-style-type: none"> • C’è evoluzione dell’evoluzione
<ul style="list-style-type: none"> • I peggiori e i medi individui tendono a non generare 	<ul style="list-style-type: none"> • I peggiori individui e i migliori tendono a non generare

Tab. 1 – Differenze tra G.A. classico e GenD.

7. Confronto tra *GenD* e Genetic Algorithm tradizionali (G.A.)

Per valutare le proprietà di *GenD* abbiamo scelto un problema classico di ottimizzazione, già affrontato tramite i G.A. tradizionali.

Si tratta di un problema-giocattolo presentato da L. Davis [Davis, 1991], in un articolo tecnico e accompagnato da un software specifico dei G.A. per affrontarlo.

Si tratta del problema dei “3 colori”: data una griglia $N \times N$, ogni cella della griglia deve essere riempita da un colore, a partire da una tavolozza di 3 colori costanti (poniamo verde, rosso e blu).

È necessario, però, impostare alcuni vincoli; questi sono:

- a. ogni cella deve aver un colore non uguale sia nella cella sopra che in quella sotto;
- b. ogni cella deve aver un colore non uguale sia nella cella destra che in quella sinistra;
- c. l’ultima colonna della griglia è contigua alla prima colonna;
- d. l’ultima riga della griglia è contigua alla prima riga.

Tali vincoli elementari rendono questo problema-giocattolo semplice da comprendere ed efficace per effettuare un test.

Abbiamo utilizzato il software che accompagna l’articolo di Davis per valutare le prove del G.A. tradizionale. Per valutare *GenD* abbiamo usato il nostro software, ma abbiamo codificato il problema e scritto la **funzione di fitness** di *GenD* in modo identico a come la codifica è stata effettuata nel software di Davis: ogni individuo è composto di $N \times N$ geni ed ogni gene ha un alfabeto di 3 caratteri, uno per colore; inoltre la fitness massima che ogni individuo può raggiungere è data da $2 \times (N \times N)$; cioè: il massimo di **non contiguità di colore per riga** ($N \times N$) ed il massimo di **non contiguità di colore per colonna** ($N \times N$).

Ad esempio: se la griglia da colorare è 5×5 , allora ogni “individuo” sarà formato da 25 geni e una fitness di 50 indicherà tutti gli individui che rispettano tutti e quattro i vincoli (a, b, c, d) che abbiamo prima esposto.

Possiamo ricordare che in una griglia 5×5 le colorazioni possibili sono 847.288.609.443, mentre le combinazioni accettabili (50/50 punti) sono 1.000 [Davis, 1991].

I criteri di valutazione che proponiamo per questo confronto sono quattro:

1. il **numero di generazioni** che il G.A. tradizionale e *GenD* impiegano per trovare la **prima soluzione** (più piccolo è, meglio è);
2. il **numero di individui** necessari per trovare **almeno una soluzione** (meno è, meglio è);
3. il **numero di generazioni** utili per trovare la prima soluzione al **crescere delle dimensioni della griglia**, tenendo costante il **numero degli individui** (poche generazioni e grandi griglie è meglio);
4. il **numero** e la **rapidità di individuazione** di soluzioni accettabili, ma diverse tra di loro, dopo la prima soluzione (più è, meglio è).

La tab. 2 riporta i risultati ottenuti in una griglia 5×5 e 100 individui. Nel G.A.

classico si è usato un tasso di mutazione dell'1% (come preimpostato da Davis per questo test).

Sono stati eseguiti otto test per ciascun algoritmo, uno di seguito all'altro, usando lo stesso generatore di numeri random e si sono scelti i sette risultati migliori di ogni algoritmo.

G.A. Classic		GenD		
<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>Soluzioni nuove per generazione</i>
1	53	1	76	4.5
2	627**	2	60	17.5
3	133	3	50	7.9
4	109	4	74	4.3
5	29*	5	81**	2
6	31	6	30*	3.7
7	134	7	53	4.9
<i>Media</i>	$\cong 145$	<i>Media</i>	$\cong 60$	$\cong 6.4$

* test migliore - ** test peggiore

Tab. 2 – Tre colori: 5×5 : 100 individui.

Nota: ogni soluzione diversa, in G.A. equivale a riattivare l'algoritmo dall'inizio. Nell'ultima colonna viene espresso il numero medio di generazioni dopo il quale GenD trova una **soluzione nuova** rispetto alle precedenti; questo test è effettuato per 100 generazioni dalla prima soluzione.

La tab. 3 mostra i risultati raggiunti dai due algoritmi con una popolazione di 50 individui. I criteri utilizzati per il test sono analoghi a quelli precedenti.

G.A. Classic		GenD		
<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>Soluzioni nuove per generazione</i>
1	156	1	84	11.6
2	158	2	91	13.6
3	549**	3	50*	5.5
4	541	4	66	10.3
5	230	5	66	3.6
6	152*	6	50*	10.7
7	284	7	129**	7.7
<i>Media</i>	$\cong 295$	<i>Media</i>	$\cong 89$	$\cong 8.9$

* test migliore - ** test peggiore

Tab. 3 - Tre colori: 5×5 : 50 individui

Nella tab. 4, il test è stato effettuato con una griglia 6×6 e 100 individui di

popolazione. La procedura è stata analoga a quella dei test precedenti.

G.A. Classic		GenD		
<i>N. Test</i>	<i>Generazioni per Prima Soluzione</i>	<i>N. Test</i>	<i>Generazioni per Prima Soluzione</i>	<i>Soluzioni Nuove per Generazione</i>
1	225	1	61	0.62
2	243	2	48	1.08
3	273**	3	61	0.95
4	147	4	42*	0.65
5	154	5	47	0.73
6	140	6	62**	0.79
7	104*	7	58	0.89
<i>Media</i>	$\cong 183$	<i>Media</i>	$\cong 54$	$\cong 0.81$

* test migliore - ** test peggiore

Tab. 4 - Tre colori: 6×6 : 100 individui.

Nella tab. 5, è stato effettuato il test con una matrice 7×7 con 200 individui. Superate le 1000 generazioni, la prova è stata considerata fallita (F).

G.A. Classic		GenD		
<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>Soluzioni nuove per generazione</i>
1	991	1	92	0.73
2	F	2	214	1.03
3	F	3	F	/
4	F	4	102	1.03
5	F	5	141	1.14
6	F	6	87*	2.1
7	510*	7	137	1.06
<i>Media</i>	$\cong 750$ (29% di capacità risolutiva)	<i>Media</i>	$\cong 128$ ($\cong 85\%$ di capacità risolutiva)	$\cong 1.18$

* test migliore

Tab. 5 - Tre colori: 7×7 : 200 individui.

Lo stesso test è stato effettuato dotando i due algoritmi di 300 individui di popolazione. Le procedure sono state analoghe a quelle degli esperimenti precedenti (vedi tab. 6).

Considerando il non efficace funzionamento del G.A. classico, si è condotto un ulteriore test, sulla stessa matrice 7×7 , dotando, però, i due algoritmi di una popolazione di 400 individui (la tab. 7 mostra i risultati).

<i>G.A. Classic</i>		<i>GenD</i>		
<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>Soluzioni nuove per generazione</i>
1	F	1	84*	0.38
2	471	2	114	0.87
3	F	3	154	1.30
4	F	4	F	/
5	F	5	122	2.02
6	F	6	117	0.95
7	F	7	255	0.66
<i>Media</i>	471 (14% di capacità risolutiva)	<i>Media</i>	$\cong 141$ ($\cong 85\%$ di capacità risolutiva)	$\cong 1.03$

* test migliore

Tab. 6 - Tre colori: 7×7 : 300 individui.

<i>G.A. Classic</i>		<i>GenD</i>		
<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>N. Test</i>	<i>Generazioni per prima soluzione</i>	<i>Soluzioni nuove per generazione</i>
1	599	1	92	0.85
2	452	2	131	2.3
3	749	3	131	1.07
4	F	4	91	0.5
5	F	5	87*	0.32
6	F	6	139**	0.54
7	341*	7	89	0.25
<i>Media</i>	$\cong 535$ (57% di capacità risolutiva)	<i>Media</i>	$\cong 108$	$\cong 0.83$

* test migliore - ** test peggiore

Tab. 7 - Tre colori: 7×7 : 400 individui.

8. GenD e il TSP

8.1. Il problema

Per misurare le capacità di modellizzazione di *GenD*, abbiamo affrontato quesiti di natura diversa, tra cui quello dell'ottimizzazione di una **sequenza non ripetitiva**. Un classico esempio di questi problemi è quello noto come *TSP* (Travelling Sales Person Problem), che appartiene alla classe dei problemi *NP* completi.

La notorietà di *TSP* ha consentito di confrontare *GenD* con altri algoritmi (Greedy e varie implementazioni di Genetic Algorithm) sullo stesso Data Base.

Dato un insieme di punti $P(x, y)$, trovare il **cammino minimo** che connette tutti i punti in un grafo hamiltoniano (in modo tale che ogni punto sia connesso a due soli altri punti e il grafo sia chiuso). Il numero di percorsi possibili cresce esponenzialmente al crescere del numero dei punti, secondo l'equazione

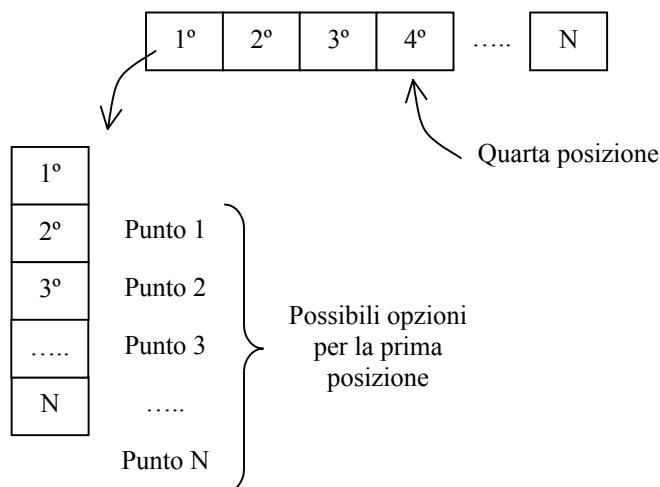
$$\text{Numero Percorsi} = \frac{(\text{Numero Punti} - 1)}{2}$$

L'algoritmo di *GenD*, per questa tipologia di problemi, è rimasto lo stesso, salvo i necessari adattamenti apportati data la natura del quesito.

8.2. La codifica del problema

Dati N punti, l'individuo è costituito da 40 geni ed ogni gene ha un alfabeto di 40 opzioni; ogni gene rappresenta la **porzione** di un punto, ed ogni opzione il punto x, y specifico in quella posizione.

Esempio:



Questa codifica è la più economica ed usata per questo tipo di problema.

Gli altri tipi di Algoritmi Genetici che sono stati confrontati con *GenD* hanno codificato il problema nello stesso modo. Ciò rende pertinente un confronto.

8.3. Gli adattamenti di *GenD*

Per adattare *GenD* alla specificità del problema sono stati effettuati i seguenti adattamenti:

- a. **Cross Over con ottimizzazione locale** (*Opportunistic Recombination*): due individui vengono accoppiati per produrre un solo offspring. L'accoppiamento avviene con una ottimizzazione locale: ogni gene viene scelto dal "padre" o dalla "madre" a seconda di quale dei due "genitori" offre la soluzione più conveniente, escludendo e scorrendo in parallelo i geni dei due genitori, partendo da un gene casuale; per il resto si tratta del tipico Cross Over previsto da *GenD*.
- b. **Mutazione**: le mutazioni vengono effettuate in modo usuale, scambiando due geni a caso dalle loro posizioni.
- c. **Operatori di Ottimizzazione Locale**: su ogni individuo si provano ad effettuare dei cambiamenti nei geni e solo se l'operazione migliora il risultato è accettata, altrimenti è ignorata.
- d. **Tribù e Contaminazioni**: questa architettura non è specifica per il problema *TSP* e può essere vista come un ulteriore sviluppo dell'algoritmo di *GenD*.
Se, per risolvere il problema, la popolazione di individui viene fissata a M , allora si possono immaginare T tribù (sottopolazioni) di $\frac{M}{T}$ individui ciascuna.
Nella fase cooperativa avviene che, ad ogni generazione, ogni tribù offre ad ogni altra il suo **migliore individuo di quella generazione ed una copia dello stesso leggermente mutata**. Ogni tribù ricevente controlla quale dei due individui offerti non è presente nella propria sottopolazione e, di conseguenza, accetta o rifiuta l'offerta.
Abbiamo definito "Contaminazione" (o anche "Immigrazione") questo operatore, in quanto consente ad ogni tribù di essere contaminata dai modelli delle altre tribù da variazioni di questi.
Parliamo quindi di **EsoMatrimonio** quando individui provenienti da 2 tribù diverse generano un figlio che andrà a sostituire nella tribù del genitore che ha il fitness più alto l'individuo con il fitness più basso.
- e. E' presente, inoltre, l'operatore di **duplicazione** che controlla in modo random tutte le sottopolazioni e, quando trova in una tribù due individui uguali, produce una mutazione casuale su uno dei due.

8.3.1. *Gli Operatori di Ottimizzazione Locale*

Gli **Operatori di Ottimizzazione Locale** intervengono ad ogni ciclo evolutivo sul singolo individuo sperimentando degli adattamenti dei geni che vengono effettuati solo se migliorano la fitness :

1. **Inversion**: l'operatore "Inversione" si serve di due geni scelti casualmente dallo stesso individuo e prova ad invertire l'ordine "prima-dopo" tra i geni selezionati.

2. **Exchange**: l'operatore “Exchange” sceglie due geni a caso (G1 e G2) (distanti tra loro di almeno 2 posizioni) e li scambia tra di loro.
3. **Swaping**: l'operatore “Swapping” sceglie un gene a caso (G1) e lo scambia con il successivo.
4. **Neighbourhood**: l'operatore di “Neighbourhood ” sceglie un gene casuale (G1), definisce un intorno (R) di quel gene, casuale anch'esso; successivamente sceglie, fuori dall'intorno il gene (G2) più vicino al primo (G1); dopodiché mette G2 in un punto dell'intorno R di G1, facendo scorrere di una posizione gli altri geni.
5. **Self-generation**: l'operatore di “Self-generation” divide i geni in tre blocchi casuali ABC, in modo tale che ogni blocco contenga più di un gene, e li riscrive in ordine BAC.
6. **Altruism**: l'operatore di “Altruism” sceglie un gene casuale (Cross1) e quindi definisce un intorno (Bound) in proporzione al numero di geni ((numGeni/4)-1). All'interno dei limiti di Bound si sceglie quindi un gene (Cross2), casuale anch'esso. Si invertono quindi Cross1 con Cross2.
7. **Shuffle**: l'operatore di “Shuffle” sceglie 3 geni casuali (Cross1, Cross2, Cross3). Sulla base della posizione occupata dai tre geni divide il genoma in 4 parti A,B,C,D e li ricombina nella sequenza B,D,C,A.
8. **Slip**: l'operatore “Slip” sceglie due geni a caso (G1 e G2), distanti tra loro di almeno 3 posizioni, e fa slittare i geni compresi tra essi di una posizione.
9. **Insertion**: l'operatore di “Insertion” sceglie due geni casuali (First e Last) che siano ad una certa vicinanza tra loro. Sceglie quindi casualmente 2 altri geni (NewFirst) e (NewLast), al di fuori dei geni compresi tra First-1(BeforeFirst) e Last-1(AfterLast). Scambia quindi tra loro newFirst con BeforeFirst e NewLast con AfterLast.
10. **Annealing**: l'operatore “Annealing” sceglie un gene a caso (G1) e quindi ridistribuisce casualmente i 5 geni che occupano le posizioni successive.

8.4. Gli esperimenti

Gli esperimenti che abbiamo condotto sono stati effettuati su punti generati casualmente dal programma “Randxy.c” di B.Reetz in “Greedy Solutions to the Travelling Sales Person Problem” Advanced Technology for Developers (ATD), volume 2, Maggio 1993, pp.8-14).

Abbiamo confrontato *GenD* con quattro algoritmi diversi:

1. “Greedy”, nella versione fornita da B. Reetz in ATD, vol. 2, maggio 1993, pp.8-14;
2. “Classic Genetic Algorithm” (CGA), nella versione software scritta da C. Klimasauskas (in ATD, vol. 2, Feb. 1993, pp.9-17)
3. “Enanced Genetic Algorithm” (EGA), nella versione scritta da C. Klimasauskas (in ATD, vol. 2, Feb. 1993, pp.9-17)

4. “Simulating Anneal” (SA), nella versione scritta da C. Klimasauskas (in ATD, vol. 2, June 1993, pp. 9-16).

I confronti sono stati eseguiti su 18 classi di punti.

E’ stata effettuata una sola prova per ogni algoritmo e la popolazioni degli algoritmi genetici è stata fissata a 50 individui, mentre nel caso di *GenD* si è lavorato con 5 tribù di 10 individui ciascuna.

8.5. I risultati

La tabella 8 riporta i risultati del confronto.

Tranne che per quello di Greedy, che ha un sistema di calcolo diverso, gli altri tre tipi di Algoritmi Genetici sono stati lasciati evolvere fino al punto che la soluzione non si stabilizzasse; è comunque stato assunto un limite di 10.000 generazioni.

Non è stato possibile effettuare confronti sulle rapidità di computazione, in quanto le tecniche software dei tre Algoritmi Genetici sono molto diverse (*GenD*, ad esempio, si serve di una grafica dinamica); l’algoritmo di Greedy ha nella rapidità il suo punto forte, per cui le tabelle dei risultati vanno lette tenendo questo dato nella dovuta importanza.

Siccome i risultati ottenuti da *GenD* sono risultati sempre i migliori, abbiamo deciso di allegare la soluzione grafica trovata da *GenD* per ogni prova (fig. 2-19).

L’indice *MST* (Minimo Spanning Tree) è stato ricavato tramite il programma MST.EXE fornito da B.Reetz in “Greedy Solutions to the Traveling sales Person Problem” in ATD, vol. 2, Maggio 1993, pp.8-14.

Nel grafico in figura 1, mostriamo le distanze (espresse in percentuali) per ogni algoritmo dall’indice *MST*.

Points Num	<i>Algorithms</i>					MST	No Optimized*	<i>Approaching in % to MST</i>				
	Greedy	G.A.	E.G.A.	S.A.	GenD			Greedy	G.A.	E.G.A.	S.A.	GenD
City 30	473	425.51	423.74	423.74	423.74	324	953	54.01	68.67	69.22	69.22	69.22
City 35	529	540.25	519.44	508.45	508.45	429	1877	76.69	74.07	78.92	81.48	81.48
City 40	570	592.34	485.23	475.29	475.29	384	1736	51.56	45.74	73.64	76.23	76.23
City 50	603.68	620.78	558.8	536.27	536.22	429	2741	59.28	55.30	69.74	75.00	75.01
City 60	696.79	771.91	624.99	624.21	619.27	494	2623	58.95	43.74	73.48	73.64	74.64
City 65	679.9	791.76	647.83	649.92	628.23	541	3326	74.33	53.65	80.25	79.87	83.92
City 70	804.92	881.67	740.26	677.7	663.96	540	3844	50.94	36.73	62.91	74.50	77.04
City 75	755.88	828	720	683.78	662.07	537	3643	59.24	45.81	65.92	72.67	76.71
City 80	730.63	880.54	716.78	702.64	673.48	549	3859	66.92	39.61	69.44	72.01	77.33
City 90	819.2	1206	806.35	793.59	748.48	606	4011	64.82	0.99	66.94	69.04	76.06
City 100	892.05	902	782	800.34	761.52	606	5502	52.80	51.16	70.96	67.93	74.34
City 120	906.34	1128.53	861.9	908.91	806.12	666	5797	63.91	30.55	70.59	63.53	77.74
Eil 51	496.4	484	437	435.45	428.87	376	1271	67.98	71.28	83.78	84.19	85.94
Eil 76	606.77	632	566.95	567.79	544.36	441	1950	62.41	56.69	71.44	71.25	76.56
Eil 101	736.36	986.55	687.89	685.75	642.03	516	2064	57.29	8.81	66.69	67.10	75.77
Berlin 52	8182.19	8618.92	7958.05	7544.36	7544.36	5988	21802	63.36	56.06	67.10	74.01	74.01
Bier 127	125023	161428.59	123062.75	123969	118562	93844	390833	66.78	27.98	68.86	67.90	73.66
Ch 130	7195.33	9425.41	6478.14	6457.17	6147.82	5072	47503	58.14	14.17	72.28	72.69	78.79

(*) Il valore riportato in colonna corrisponde alla media di N prove di calcolo random della distanza di un set di punti.

Tab. 8 – Risultati del confronto di alcuni algoritmi sul problema TSP.

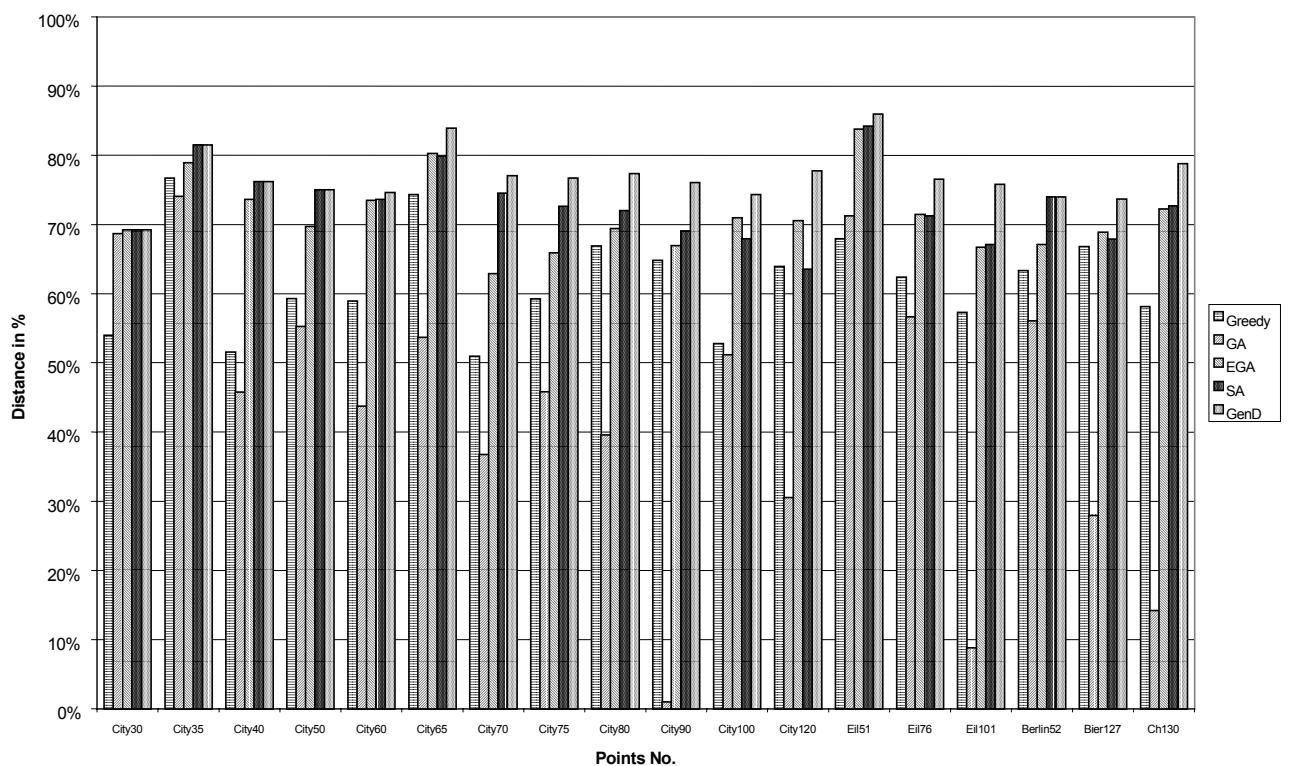


Fig. 1 - Grafico delle distanze in % di ogni algoritmo dall'indice MST

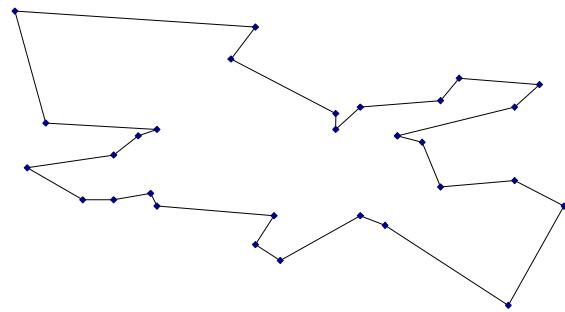


Fig. 2 – Soluzione grafica trovata da GenD su 30 punti (323.74).

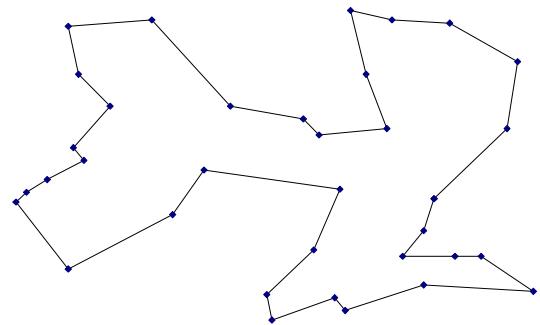


Fig. 3 – Soluzione grafica trovata da GenD su 35 punti (508.45).

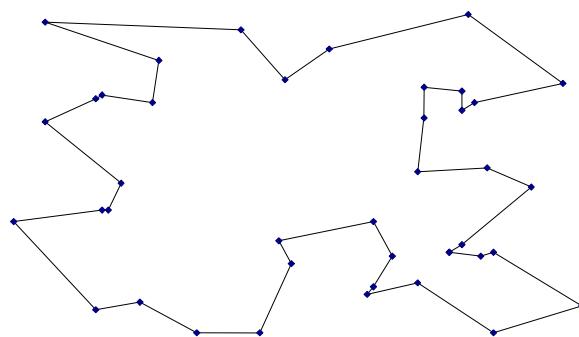


Fig. 4 – Soluzione grafica trovata da GenD su 40 punti (475.29).

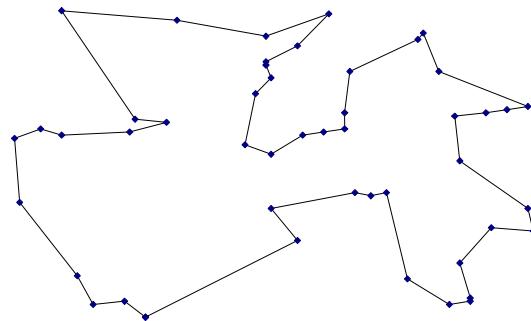


Fig. 5 – Soluzione grafica trovata da GenD su 50 punti (536.27).

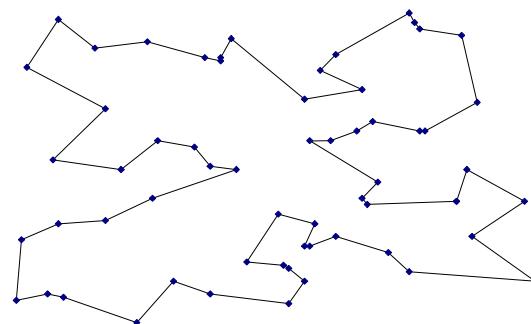


Fig. 6 – Soluzione grafica trovata da GenD su 60 punti (619.27).

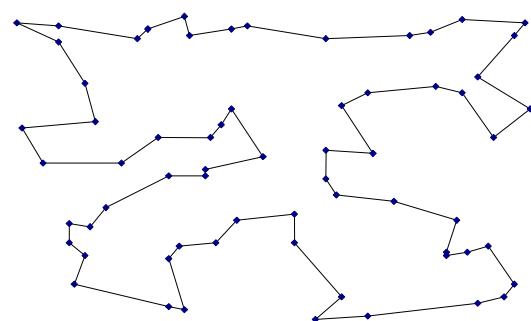


Fig. 7 – Soluzione grafica trovata da GenD su 65 punti (628.23).

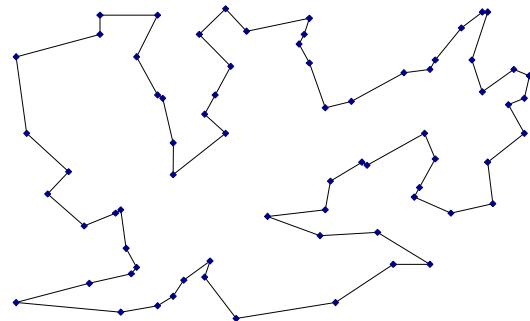


Fig. 8 – Soluzione grafica trovata da GenD su 70 punti (663.96).

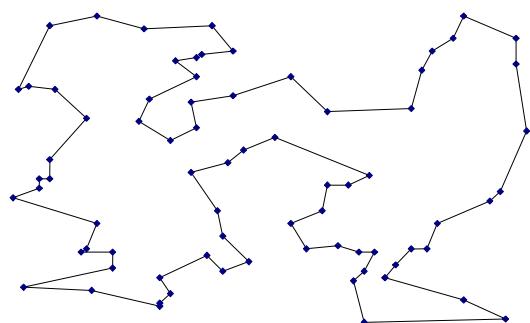


Fig. 9 – Soluzione grafica trovata da GenD su 75 punti (662.07).

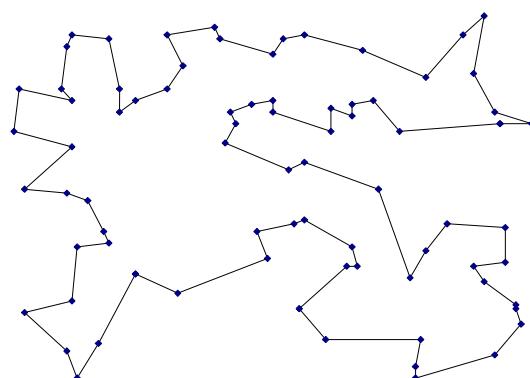


Fig. 10 – Soluzione grafica trovata da GenD su 80 punti (673.48).

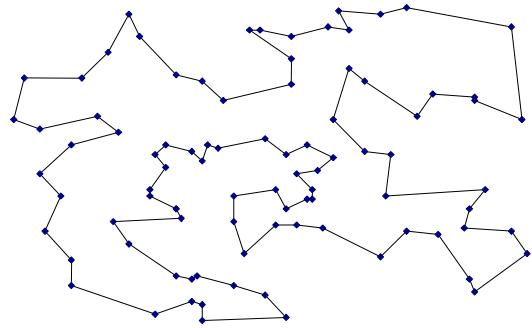


Fig. 11 – Soluzione grafica trovata da GenD su 90 punti (748.48).

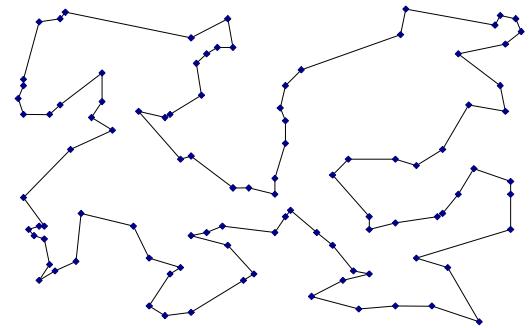


Fig. 12 – Soluzione grafica trovata da GenD su 100 punti (761.52).

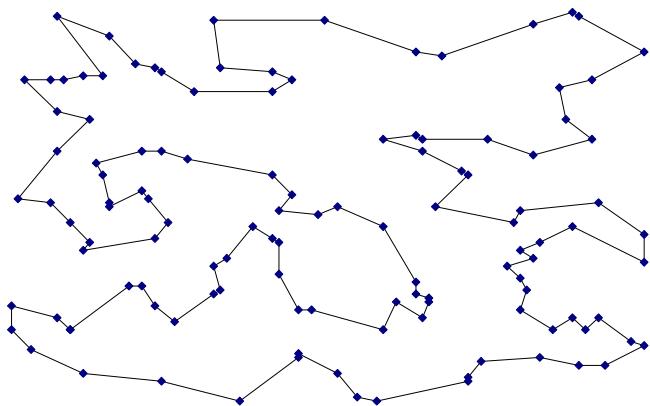


Fig. 13 – Soluzione grafica trovata da GenD su 120 punti (806.12).

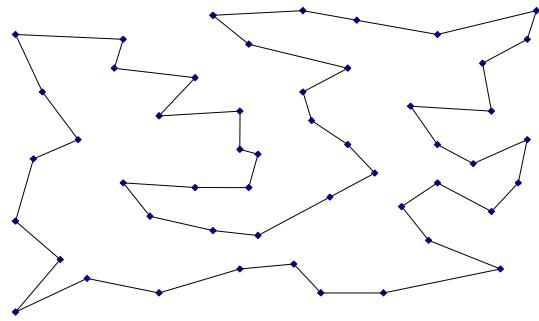


Fig. 14 – Soluzione grafica trovata da GenD su 51 punti (428.87).

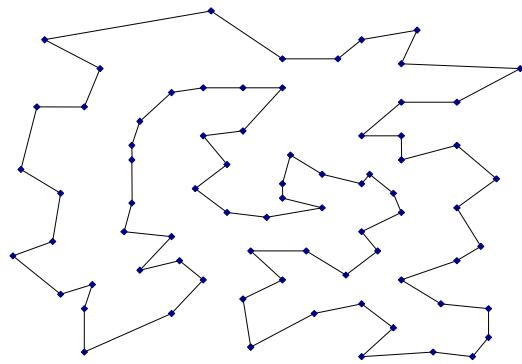


Fig. 15 – Soluzione grafica trovata da GenD su 76 punti (544.36).

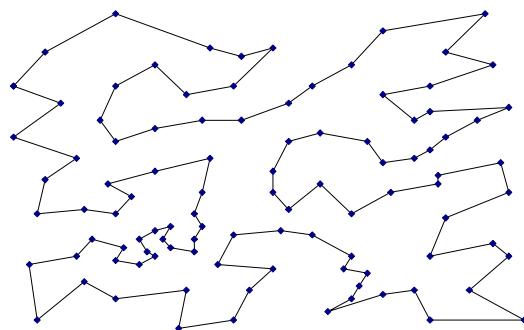


Fig. 16 – Soluzione grafica trovata da GenD su 101 punti (642.03).

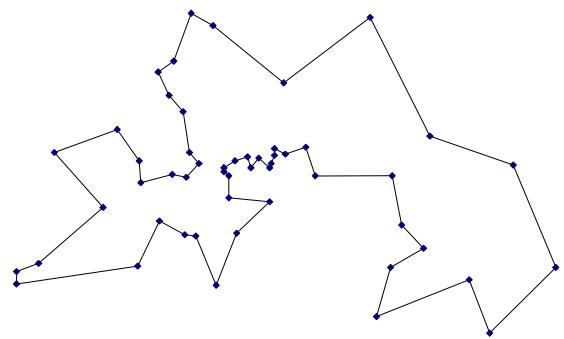


Fig. 17 – Soluzione grafica trovata da GenD su 52 punti (7544.36).

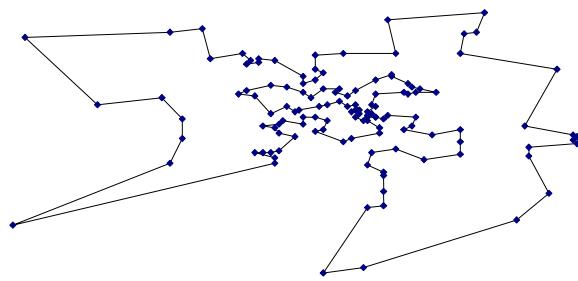


Fig. 18 – Soluzione grafica trovata da GenD su 127 punti (118562).

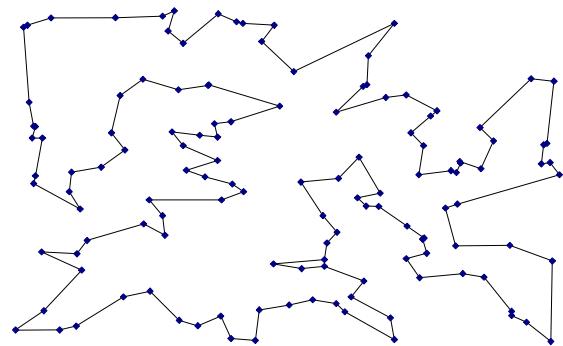
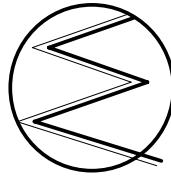


Fig. 19 – Soluzione grafica trovata da GenD su 130 punti (6147.82).

Riferimenti bibliografici

- Ackley D. H. (1987), *Connectionist Machine for Genetic Hillclimbing*, Kluwer, Boston, 1987.
- Arbib M. A. (1995), (Edited by) *The Handbook of Brain Theory and Neural Networks*, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 1995.
- Bovet D. P. and P. L. Crescenzi (1992), *Teoria della Complessità Computazionale*, Franco Angeli Editore, Milano, 1992 (*Theory of Computational Complexity*).
- Buscema M. & Semeion Group (1999a), *Reti Neurali Artificiali e Sistemi Sociali Complessi*, Vol. I: *Teoria e Modelli*, Franco Angeli, Milano, 1999.
- Buscema M. (1999b), “Genetic Doping Algorithm”, in BUSCEMA 1999a, pp. 465–486.
- Caudil H. and Ch. Butler (1990), *Naturally Intelligent Systems*, MIT Press, Cambridge, MA, 1990.
- Chalmers D. J. 1990, “The Evolution of Learning: An Experiment in Genetic Connectionism”, in TOURETZKY 1990b.
- Dagli C. H. and S. Sittisathanchai (1993), “Genetic Neuroschedular for Job Shop Scheduling”, *Computers & Industrial Engineering*, 25, 267-270, 1993.
- Davis L. (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, ITP, 1991.
- Dawid H. (1999), *Adaptive Learning by Genetic Algorithms. Analytical Results and Applications to Economic Models*, Springer-Verlag, Heidelberg, New York, 1999.
- Deboeck T. and G. J. Deboeck (1992), “Genetic Optimization of Neural Nets for Trading”, *ATD*, vol. 1, October 1992, pg. 1-9.
- Eldredge N. (1995), *The Great Debate at the High Table of Evolutionary Theory*, John Wiley & Sons, New York, 1995 (in italiano: Einaudi, Turin, 1998).
- Friedrich C. M. and C. Moraga (1997), “Using genetic Engineering to Find Modular Structures and Activation Functions for Architectures of Artificial Neural Networks”, *Computation Intelligence, Theory and Applications, Proceedings of the 5th Fuzzy Days*, (LCNS 1226), 150–161, Dortmund, Germany.
- Gen M. and R. Cheng (1997), *Genetic Algorithms and Engineering Design*, Wiley Interscience, New York, 1997.
- Harp S. A., T. Samad and A. Guha (1990), “Designing Application-Specific Neural Networks Using the Genetic Algorithm”, in TOURETZKY 1990a.
- Kohonen T. (1995), *Self-Organizing Maps*, Springer Verlag, Berlin, Heidelberg, 1995.
- Kosko B. (1993), *Fuzzy Thinking. The New Science of Fuzzy Logic*, Hyperion/Disney Books, 1993.

- Koza J. R. (1992), *Genetic Programming On the Programming of Computers by Means of Natural Selection*, A Bradford Book, The MIT Press, Cambridge, Massachusetts London, England, 1992.
- Miller G. F. and P. M. Todd (1990), "Exploring Adaptive Agency I: Theory and Methods for Simulating the Evolution of Learning", in TOURETZKY 1990a.
- Mitchell M. (1996), *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, MA, 1996.
- Nolfi S. (1995), "'Genotypes" for Neural Networks", in ARBIB 1995.
- Quagliarella D., J. Periaux, C. Polani and G. Winter (eds.) (1998), *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, John Wiley and Sons, Ltd, England, 1998.
- Rao S. S. (1996), *Engineering Optimization. Theory and Practice*, Wiley Interscience, New York, 1996.
- Rawling G. (1991), *Foundations of Genetic Algorithms*, Morgan Kaufman, San Mateo, CA., 1991.
- Rogers D. (1990), "Predicting Weather Using a Genetic Memory: A Combination of Kanerva's Sparse Distributed Memory with Holland's Genetic Algorithms", in TOURETZKY 1990a.
- Rumelhart D. E. and J. L. McClelland (1986), (eds.) *Parallel Distributed Processing, Vol.1 Foundations, Explorations in the Microstructure of Cognition, Vol.2 Psychological and Biological Models*. The MIT Press, Cambridge, MA, London, England 1986.
- Touretzky D. S. (ed) (1990a), *Advances in Neural Information Processing Systems*, vol. 2, Morgan Kaufman, San Mateo, CA, 1990.
- Touretzky D. S., (ed.) (1990b), *Connectionist Models, Proceedings of the 1990 Summer School*, Morgan Kaufman, San Mateo, CA, 1990.
- Whitley L. D. (ed.) (1993), *Foundation of Genetic Algorithms 2*, Morgan Kaufmann, San Mateo, CA, 1993.
- Whitley L. D. and M. D. Vose (eds.), (1995) *Foundation of Genetic Algorithms 3*, Morgan Kaufmann, San Mateo, CA, 1995.



SEMEION
CENTRO RICERCHE DI SCIENZE DELLA COMUNICAZIONE

Semeion Technical Papers

- | | | | |
|--------------|--|--------------|---|
| P 1-2 | M. BUSCEMA, G. MASSINI, <i>L'immagine della criminalità nei Mass Media, La Criminalità nei Telefilm</i> , (vers. Ita/Ingl). 1984 | TP 12 | M. BUSCEMA, <i>Dalle Macchine alle Reti</i> , (vers. Ita). 1992 |
| TP 3 | M. BUSCEMA, <i>Dinamiche Violente</i> , (vers. Ita/Ingl). 1985 | TP 13 | M. BUSCEMA, <i>Teoria del Disagio in Incubazione</i> , (vers. Ita/Ingl). 1994 |
| TP 4 | M. BUSCEMA, <i>Teoria della Procedura di Prevenzione</i> , (vers. Ita/Ingl). 1985 | TP 14 | M. BUSCEMA, <i>Self-Reflexive Networks. Theory, Topology, Applications</i> , (vers. Ingl). 1994 |
| TP 5 | M. BUSCEMA, <i>Introduzione al Modello MQ</i> , (vers. Ita/Ingl/Franc). 1987 | TP 15 | M. BUSCEMA, <i>MAC.P.: Modello di Autorganizzazione Cognitiva in Parallello</i> , (vers. Ita). 1995 |
| TP 6 | M. BUSCEMA, <i>Intelligenza Artificiale: Sistemi Esperti e Sistemi Complessi</i> , (vers. Ita). 1987 | TP 16 | G. DIDONÉ, <i>Studio Dinamico della Rete AutoRiflessiva Monodedicata</i> , (vers. Ita). 1994 |
| TP 7 | M. BUSCEMA, <i>Il panorama teorico del Mac P</i> , (vers. Ita/Ingl). 1988 | TP 17 | M. PANDIN, <i>Analisi del Comportamento della Rete AutoRiflessiva</i> , (vers. Ita). 1994 |
| TP 8 | M. BUSCEMA, G. MASSINI, L. CASSARDO, <i>Il Modello G.T.: Modello per lo studio dell'Ergonomia dell'Interfaccia Uomo/Macchina</i> , (vers. Ita). 1988 | TP 18 | M. BUSCEMA, <i>Reti Neurali Constraint Satisfaction e Recirculation</i> , (vers. Ita/Ingl). 1995 |
| TP 9 | M. BUSCEMA, <i>Il Cervello nel Cervello</i> , (vers. Ita/Ingl/Franc). 1989 | TP 19 | M. BREDA, <i>Basic Principles on Back-Propagation Neural Networks</i> , (vers. Ingl). 1997 |
| TP 10 | M. BUSCEMA, L. CASSARDO, <i>Osservatorio Permanente sul Disagio in Incubazione</i> , (vers. Ita/Ingl/Franc). 1990 | TP 20 | M. BUSCEMA, <i>Genetic Doping and PST: a New Mapping System</i> , (vers. Ingl). 2001 |
| TP 11 | M. BUSCEMA, <i>Codici e Linguaggi: Introduzione</i> , (vers. Ita). 1991 | TP 21 | M. BUSCEMA, M. BREDA, S. TERZI, <i>Sine Net</i> , (vers. Ita). 2000-2002 |

Via Sersale 117/119, 00128 ROMA ~ Tel. 06/56652350 ~ Fax
Internet - <http://www.semeion.it> ~ E-mail: semeion@semeion.it
Ente scientifico e morale con personalità giuridica riconosciuta dallo Stato Italiano tramite
il Ministero dell'Università e della Ricerca Scientifica e Tecnologica con decreto
ministeriale del 12 novembre 1991, Registro 15 Università e Ricerca, foglio n° 269
Iscrizione Tribunale 2174 Ufficio Provvedimenti Speciali