

Calcolo Enigmatico risolto con Algoritmi Genetici

Progetto per l'esame di Sistemi Intelligenti

Luca Guerra
Matricola 791528

Indice

1	Problema	2
1.1	Descrizione	2
1.2	Scopo del lavoro	2
2	Algoritmo genetico	3
2.1	Struttura	3
2.2	Codifica degli individui	3
2.3	Funzione di fitness	4
2.4	Operatore di selezione ed elitismo	4
2.5	Operazione di crossover	5
2.6	Operazione di mutazione	5
2.7	Convergenza a ottimo locale	5
3	Software	7
3.1	Tecnologie utilizzate e dipendenze	7
3.2	Avvio e configurazione	8
3.3	Uso	8
A	Problemi e soluzioni di esempio	9

1 Problema

1.1 Descrizione

Calcolo enigmatico è il nome dato dalla nota rivista italiana “*La Settimana Enigmistica*” a una particolare categoria di problemi logico-matematici.

Un quesito di *calcolo enigmatico* si presenta come un insieme di semplici operazioni aritmetiche tra numeri interi in cui a ogni cifra è stata sostituita un simbolo in modo tale che a simbolo uguale corrisponda cifra uguale. Per esempio, il problema che compare nel numero 8656 del settimanale è il seguente:

$$\begin{array}{rcccc} AB & \times & C & = & DEB \\ + & & - & & - \\ FGB & : & H & = & HA \\ \hline DIB & - & A & = & DBE \end{array}$$

Figura 1: Esempio di problema di calcolo enigmatico

Il problema definisce sei equazioni. Tre di queste sono leggibili in orizzontale:

$$\begin{array}{l} AB \times C = DEB \\ FGB : H = HA \\ DIB - A = DBE \end{array}$$

E le altre tre in verticale:

$$\begin{array}{l} AB + FGB = DIB \\ C - H = A \\ DEB - HA = DBE \end{array}$$

Per risolverlo, occorre associare a ogni lettera una cifra diversa in modo tale che tutte le equazioni risultino soddisfatte. Per esempio, in questo caso, la sostituzione

$$A \rightarrow 4, B \rightarrow 0, C \rightarrow 9, D \rightarrow 3, E \rightarrow 6, F \rightarrow 2, G \rightarrow 7, H \rightarrow 5, I \rightarrow 1$$

produce il seguente schema:

$$\begin{array}{rcccc} 40 & \times & 9 & = & 360 \\ + & & - & & - \\ 270 & : & 5 & = & 54 \\ \hline 310 & - & 4 & = & 306 \end{array}$$

Come si può facilmente verificare tutte le operazioni sono corrette e dunque il problema è risolto.

1.2 Scopo del lavoro

In questo progetto si desidera valutare l'efficienza di algoritmi genetici per la risoluzione problemi di questo tipo, senza utilizzare metodi algebrici o di *constraint solving*.

2 Algoritmo genetico

Si è prodotto un programma che implementa un algoritmo genetico per risolvere problemi di questo tipo.

2.1 Struttura

L'algoritmo segue il normale workflow degli algoritmi genetici, con alcune variazioni.

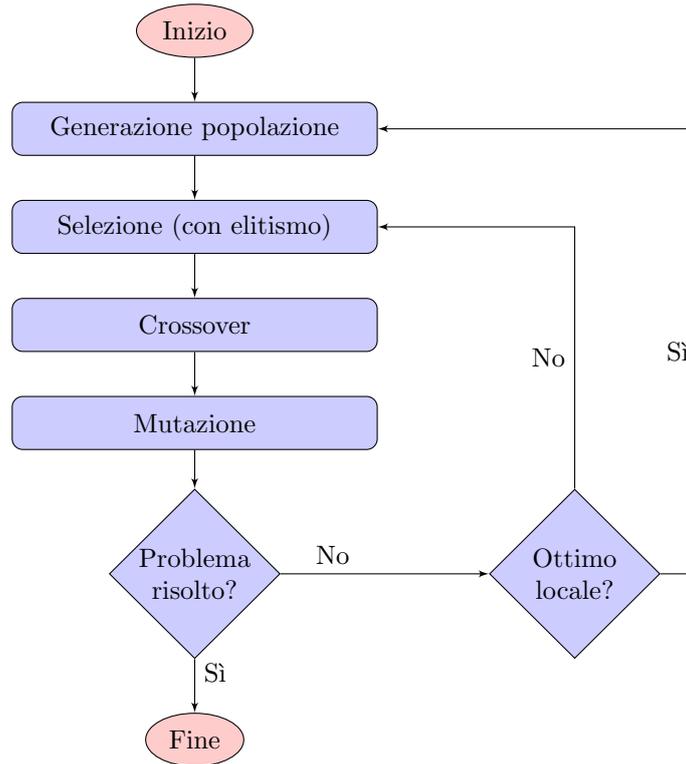


Figura 2: Schema dell'algoritmo

In particolare, è stata introdotta una funzione di riconoscimento dei casi di convergenza a ottimo locale, possibili in questo genere di problema. In tal caso, l'algoritmo provvede a introdurre nuovi elementi (generati casualmente) nella popolazione sostituendo in tutto o in parte la popolazione esistente, che a quel punto è costituita da individui molto simili che rappresentano piccole variazioni dell'ottimo locale raggiunto.

2.2 Codifica degli individui

A seguito della descrizione, è evidente che lo spazio delle soluzioni di un problema di calcolo enigmatico è l'insieme delle permutazioni delle 10 cifre numeriche. Per questo motivo si è deciso di codificare il singolo individuo semplicemente come tale permutazione. A ogni posizione del vettore della permutazione viene fatta dunque corrispondere una e una sola lettera presente nella formulazione del problema. Per esempio, la soluzione del problema in Figura 1 e il relativo mapping risultano essere:

A	B	C	D	E	F	G	H	I	
4	0	9	3	6	2	7	5	1	8

La cardinalità dello spazio degli individui risulta quindi essere $10! = 3628800$.

2.3 Funzione di fitness

La funzione di fitness scelta per questo problema è così articolata, dato un qualsiasi individuo:

- Si applica la permutazione associata all'individuo a tutte e sei le equazioni del problema, ottenendo dunque sei assegnamenti numerici.
- Per ogni assegnamento, si considera la parte prima del segno = e si calcola il risultato dell'espressione rappresentata.
- Si confronta il risultato così ottenuto con la parte dopo il segno = dell'assegnamento. In particolare, si confrontano una per una le cifre ottenute. Se una cifra del risultato calcolato è uguale alla cifra corrispondente del numero dopo il segno =, il valore di fitness dell'individuo è incrementato di 1.

Per esempio, considerando sempre il problema di Figura 1 e il seguente individuo:

A	B	C	D	E	F	G	H	I	
4	0	9	6	3	2	7	5	1	8

Si esamina la prima equazione:

$$AB \times C = DEB$$

Si sostituiscono le lettere con cifre

$$40 \times 9 = 630$$

A questo punto si calcola 40×9 , che fornisce 360 come risultato, e lo si confronta con il valore che risulta dalla sostituzione (630)

3	6	0
6	3	0

Una sola cifra corrisponde, dunque il valore di fitness di questo individuo rispetto alla prima equazione è 1. Si procede in questo modo per le altre cinque equazioni e si ottiene fitness complessiva 8.

Si nota immediatamente che la fitness massima di un individuo, cioè la fitness della soluzione del problema, è la somma della lunghezza in cifre di tutti i risultati delle equazioni (la parte destra di ogni assegnamento). Nell'esempio proposto tale valore è 15. L'operazione "Problema risolto?" dello schema in Figura 2 si riduce quindi a un confronto della fitness di ogni individuo con la fitness della soluzione.

Casi particolari In alcuni casi non è possibile applicare direttamente la funzione di fitness per un'equazione. Questo può succedere quando l'operazione è una divisione per zero, fornisce un risultato non intero, negativo, un risultato la cui lunghezza non è la stessa con il valore sostituito oppure quando l'operazione coinvolge numeri di più di una cifra che iniziano per 0. In questi casi normalmente la fitness dell'equazione per quell'individuo viene posta automaticamente a 0.

2.4 Operatore di selezione ed elitismo

Si è scelto di utilizzare il classico operatore di selezione *fitness proportional* tramite roulette. Prima dell'operazione di selezione vengono inoltre riservati due slot per gli individui con fitness maggiore della generazione precedente che vengono mantenuti senza crossover.

2.5 Operazione di crossover

L'operazione di crossover com'è normalmente definita non è adatta al caso in cui l'individuo rappresenti una permutazione. Si consideri, per esempio, un'operazione di *single point crossover* attorno al centro del vettore, tra i seguenti due individui:

$$\begin{array}{cccccc|cccc} 4 & 0 & 9 & 6 & 3 & & 2 & 7 & 5 & 1 & 8 \\ 3 & 7 & 8 & 5 & 1 & & 6 & 2 & 0 & 4 & 9 \\ \hline 4 & 0 & 9 & 6 & 3 & & 6 & 2 & 0 & 4 & 9 \end{array}$$

Appare evidente che il risultato non è una permutazione valida. Presenta infatti le cifre 4, 0, 9 e 6 ripetute due volte e non contiene le cifre 7, 8, 5, e 1. Si potrebbe pensare di scartare i risultati delle operazioni di crossover che non rappresentino una permutazione, ma questo produrrebbe un numero eccessivo di individui non validi e renderebbe l'algoritmo estremamente inefficiente.

Al contrario, in letteratura sono noti vari algoritmi di crossover che si applicano alle permutazioni. Dopo alcuni prototipi e verifiche si è deciso di implementare una variante dell'*uniform crossover* adattato al caso delle permutazioni.

Dati due individui p_1 e p_2 visti come vettori di di dimensione 10 e un valore di probabilità P_c si generano due discendenti off_1 e off_2 corrispondenti ai due individui di partenza e si procede come segue:

- Per ogni posizione i da 1 a 10, l'operatore ha effetto sulla posizione i con probabilità P_c .
- Se l'operatore ha effetto, si scambiano in off_1 e off_2 le posizioni dei valori corrispondenti a $off_1[i]$ e $off_2[i]$.

Per esempio, partendo dai due individui dell'esempio precedente e operando uno scambio in posizione 2 (evidenziata) si ottengono due nuovi individui effettuando le modifiche marcate in grassetto:

$$\begin{array}{cc|c|cccccccc} p_1 & 4 & 0 & 9 & 6 & 3 & 2 & 7 & 5 & 1 & 8 \\ p_2 & 3 & 7 & 8 & 5 & 1 & 6 & 2 & 0 & 4 & 9 \\ \hline off_1 & 4 & \mathbf{7} & 9 & 6 & 3 & 2 & \mathbf{0} & 5 & 1 & 8 \\ off_2 & 3 & \mathbf{0} & 8 & 5 & 1 & 6 & 2 & \mathbf{7} & 4 & 9 \end{array}$$

2.6 Operazione di mutazione

Dopo aver generato ciascun individuo della nuova generazione, può essere eseguito l'operazione di mutazione con probabilità P_m . In tal caso viene effettuato uno scambio tra due posizioni casuali del vettore che rappresenta l'individuo.

2.7 Convergenza a ottimo locale

Implementando il sistema descritto nei paragrafi precedenti si è verificato che l'algoritmo tende a convergere piuttosto rapidamente (come mostrato nel grafico in Figura 3, dove f_{max} è la massima fitness presente nella generazione e f_{avg} è la fitness media della generazione), ma in alcuni casi non riesce a raggiungere la soluzione. Riesce infatti a trovare un insieme di individui che si distinguono per un buon valore di fitness, facendo sì che le generazioni successive siano inondate da individui simili, che però non rappresentano la soluzione al problema e non sono nemmeno sufficientemente *vicini* alla soluzione per far raggiungere l'obiettivo tramite gli operatori di selezione, crossover e mutazione. In quel caso l'algoritmo difficilmente riuscirà a raggiungere l'ottimo globale e rimarrà bloccato indefinitamente. Tale comportamento, noto in letteratura come *local optimum convergence*, è evidenziato nel grafico in Figura 4 .

In alcuni generi di problemi questo comportamento è accettabile, ma non in questo caso, visto che l'obiettivo è quello di trovare una soluzione con fitness massima. In letteratura sono note diverse strategie che consentono di mantenere una certa diversità genetica garantita all'interno di

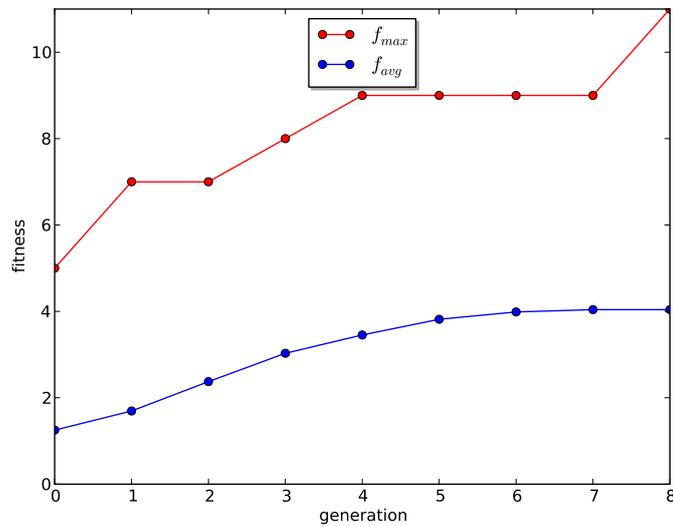


Figura 3: Convergenza rapida alla soluzione ($fit = 11$)

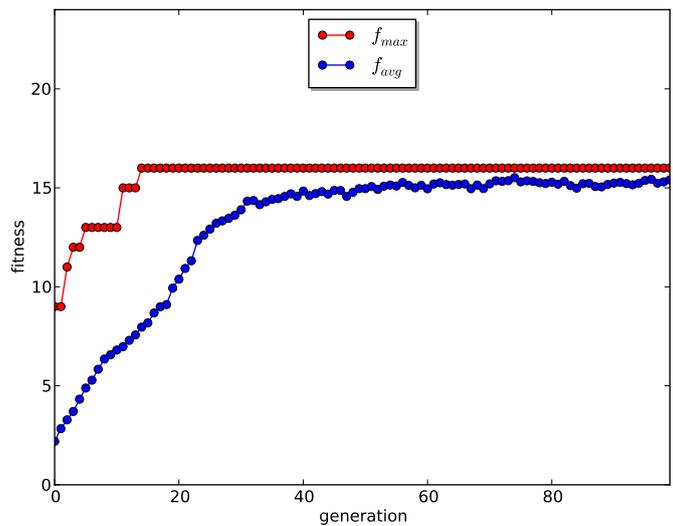


Figura 4: Convergenza a ottimo locale

ogni generazione, consentendo di dare all’algoritmo meno probabilità di rimanere bloccato in un ottimo locale al costo di avere una convergenza più lenta verso la soluzione.

Per questo problema invece si è pensato di utilizzare una strategia differente: come già accennato si è preferito implementare un meccanismo per identificare quando l’algoritmo risulta bloccato in una situazione del genere e procedere dunque a sostituire in tutto o in parte la popolazione della generazione corrente con nuovi individui.

Per ottenere tale effetto si è deciso di osservare la variazione della fitness media da una generazione all’altra. Se l’algoritmo sta raggiungendo un ottimo (globale o locale) la fitness media sarà osservata in costante aumento. Altrimenti, la fitness media rimarrà pressoché invariata.

Detta $f_{avg}(i)$ la fitness media degli individui della generazione i , a ogni nuova generazione j l'algoritmo calcolerà $\Delta = f_{avg}(j) - f_{avg}(j - 1)$ e la introdurrà in una coda di lunghezza fissa n di cui, a partire dalla generazione n -esima, calcolerà la media Δ_m . Questo procedimento è noto come media mobile (*moving average*). Se Δ_m risulta inferiore di una threshold fissata, l'algoritmo assumerà di avere raggiunto un ottimo locale e si comporterà di conseguenza. Il grafico in Figura 5 mostra un caso di convergenza a ottimo locale identificato e risolto. In questo caso, per evidenziare il comportamento sul grafico, l'algoritmo è ripartito dallo stato iniziale non appena ha identificato la convergenza.

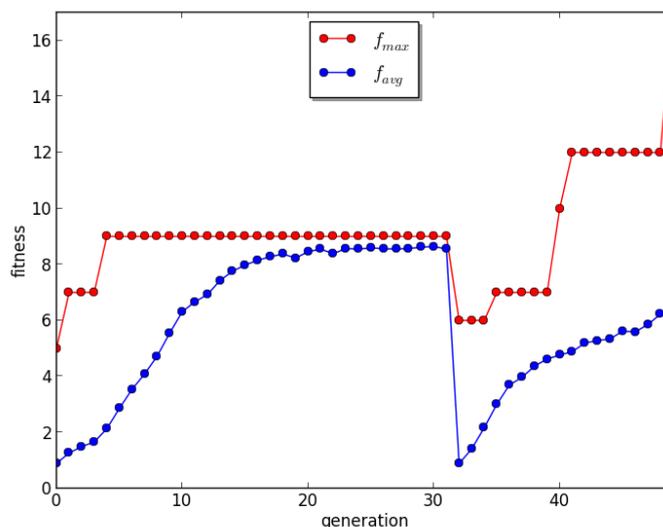


Figura 5: Convergenza a ottimo locale risolta e completata con convergenza alla soluzione (17)

3 Software

Di seguito si presenta il software realizzato implementando l'algoritmo appena esposto.

3.1 Tecnologie utilizzate e dipendenze

Il programma è scritto in Python e necessita della versione 2.7, di cui le recenti versioni di distribuzioni Linux e Mac OSX sono provviste nativamente. Il software presenta un'interfaccia grafica realizzata con le librerie PySide, basate sul framework Qt. PySide non è normalmente presente di default sulle distribuzioni Linux e Mac OSX e deve dunque essere installato a parte, dal sito ufficiale <http://qt-project.org/wiki/Category:LanguageBindings::PySide::Downloads> o tramite il sistema di gestione dei pacchetti del proprio sistema operativo. A quel punto è sufficiente eseguire `python gui.py` nella directory dei sorgenti del programma.

Microsoft Windows non è normalmente provvisto né dell'interprete Python né delle librerie necessarie, per questo si fornisce insieme al sorgente l'applicazione precompilata per Windows comprensiva di una versione leggera dell'interprete Python e tutte le dipendenze `.dll`. Per avviarla è sufficiente scompattare l'archivio `calcolo_enigmatico_win32bin.zip` ed eseguire `calcolo_enigmatico.exe`.

3.2 Avvio e configurazione

Una volta avviato, il programma si presenta così:

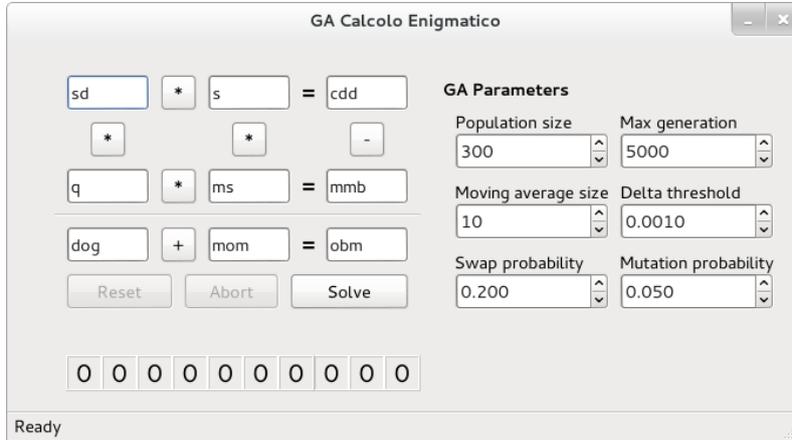


Figura 6: Programma in standby

La parte destra della finestra mostra i parametri dell'applicazione. In ordine, la dimensione della popolazione, la generazione massima (raggiungerla provoca l'arresto dell'algorithm), la dimensione n della media mobile discussa nel paragrafo 2.7 e la relativa threshold, la probabilità P_c dell'operatore di crossover discussa nel paragrafo 2.5 e la probabilità di mutazione P_m discussa in 2.6.

3.3 Uso

La parte sinistra della finestra mostra un problema di test pre-configurato e consente di introdurre qualunque problema di questo tipo, inserendo i valori nelle caselle e premendo i bottoni per cambiare gli operatori. Per avviare l'algorithm è sufficiente premere **Solve**. Durante l'esecuzione il programma mostra l'individuo con maggior fitness della popolazione più recente. Una volta risolto il problema, viene mostrato il mapping tra lettere e cifre nella parte inferiore della finestra e i numeri vengono mostrati al posto delle lettere. Per ripristinare la formulazione del problema è sufficiente premere **Reset**.

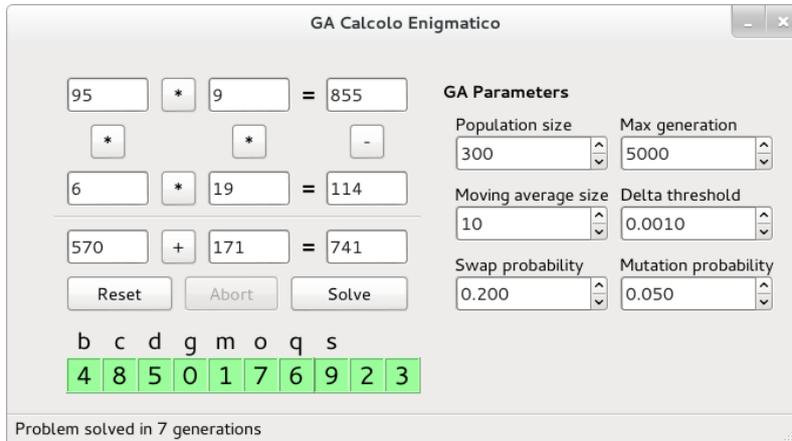


Figura 7: Problema risolto

A Problemi e soluzioni di esempio

$\begin{array}{rclcl} \text{PLS} & + & \text{LEK} & = & \text{MML} \\ & & \times & & \times \\ \text{LS} & + & \text{P} & = & \text{LR} \\ \hline \text{EL} & \times & \text{PGR} & = & \text{GQPR} \end{array}$	$\begin{array}{rclcl} 315 & + & 126 & = & 441 \\ & & \times & & \times \\ 15 & + & 3 & = & 18 \\ \hline 21 & \times & 378 & = & 7938 \end{array}$
$\begin{array}{rclcl} \text{HHI} & + & \text{CMC} & = & \text{BQH} \\ & & \times & & \times \\ \text{B} & - & \text{C} & = & \text{J} \\ \hline \text{MHJ} & + & \text{COQ} & = & \text{HAQJ} \end{array}$	$\begin{array}{rclcl} 116 & + & 585 & = & 701 \\ & & \times & & \times \\ 7 & - & 5 & = & 2 \\ \hline 812 & + & 590 & = & 1402 \end{array}$
$\begin{array}{rclcl} \text{G} & \times & \text{AJ} & = & \text{SKA} \\ & & \times & & \times \\ \text{KD} & : & \text{GD} & = & \text{N} \\ \hline \text{SGD} & - & \text{KJ} & = & \text{OK} \end{array}$	$\begin{array}{rclcl} 2 & \times & 84 & = & 168 \\ & & \times & & \times \\ 60 & : & 20 & = & 3 \\ \hline 120 & - & 64 & = & 56 \end{array}$
$\begin{array}{rclcl} \text{FA} & - & \text{P} & = & \text{K} \\ & & \times & & \times \\ \text{ND} & : & \text{AK} & = & \text{A} \\ \hline \text{DCA} & : & \text{FFA} & = & \text{D} \end{array}$	$\begin{array}{rclcl} 12 & - & 4 & = & 8 \\ & & \times & & \times \\ 56 & : & 28 & = & 2 \\ \hline 672 & : & 112 & = & 6 \end{array}$
$\begin{array}{rclcl} \text{OA} & \times & \text{CD} & = & \text{ABD} \\ & & \times & & \times \\ \text{CM} & \times & \text{M} & = & \text{MB} \\ \hline \text{BOM} & : & \text{CA} & = & \text{MJ} \end{array}$	$\begin{array}{rclcl} 36 & \times & 18 & = & 648 \\ & & \times & & \times \\ 12 & \times & 2 & = & 24 \\ \hline 432 & : & 16 & = & 27 \end{array}$
$\begin{array}{rclcl} \text{AS} & \times & \text{AAL} & = & \text{AALS} \\ & & \times & & \times \\ \text{ASS} & + & \text{P} & = & \text{ASP} \\ \hline \text{ASSS} & + & \text{MJ} & = & \text{ASMJ} \end{array}$	$\begin{array}{rclcl} 10 & \times & 116 & = & 1160 \\ & & \times & & \times \\ 100 & + & 2 & = & 102 \\ \hline 1000 & + & 58 & = & 1058 \end{array}$
$\begin{array}{rclcl} \text{QLI} & + & \text{HPI} & = & \text{SSJ} \\ & & - & & - \\ \text{JSJ} & - & \text{JAA} & = & \text{Q} \\ \hline \text{LJQ} & \times & \text{J} & = & \text{JQA} \end{array}$	$\begin{array}{rclcl} 416 & + & 576 & = & 992 \\ & & - & & - \\ 292 & - & 288 & = & 4 \\ \hline 124 & \times & 2 & = & 248 \end{array}$
$\begin{array}{rclcl} \text{OKKK} & + & \text{K} & = & \text{OKRD} \\ & & - & & - \\ \text{MLFO} & - & \text{CFFD} & = & \text{DLFD} \\ \hline \text{CCKD} & + & \text{KFCD} & = & \text{RCRO} \end{array}$	$\begin{array}{rclcl} 4666 & + & 6 & = & 4672 \\ & & - & & - \\ 3504 & - & 1002 & = & 2502 \\ \hline 1162 & + & 6012 & = & 7174 \end{array}$