USING GENETIC ALGORITHMS TO BREAK A KNAPSACK BASED ON MULTIPLE KNAPSACKS[+]

Safaa S. Omran[*] Ali S. Al-Khalid[*] Israa F. Ali[**]

Abstract:

The genetic algorithm is one of the search methods, which finds the optimal solution. This work focuses on using Genetic Algorithms to cryptanalyse knapsack cipher based on three knapsacks. The knapsack cipher is with a knapsack sequence of size 8. Different values of parameters have been used: Population size, number of generation.

Keywords: **Genetic Algorithm,knapsack cipher,cryptanalysis.**

إستخدام الخوارزمية الجينية لفك شفرة نابساك المعتمدة على عدةknapsack

صفاء صاحب عمران علي شاكر الخالد إسراء فيصل علي

المستخلص:

الخوارزميات الجينية هي طريقة من طرق البحث التي تبحث عن الحل الأمثل. تعتبر الخوارزمية الجينية واحد من الطرق التي تستخدم لفك التشفير. يركز هذا العمل على إستخدام الخوارزمية الجينية لفك شفرة حقيبة التشفير المعتمدة على ثلاث حقائب. تم إستخدام قيم مختلفة للمتغيرات التالية: حجم المجتمع وعدد التوليد.

Introduction:

With more and more development in the field of computer networks and internet, the need for network, computer and information security is also increasing. There are different ways to secure information passed over the network. One such a technique is cryptology. Cryptology is the science and study of systems for secret communication [1]. Cryptography is the science of building new powerful and efficient encryption and decryption methods. It deals with the techniques for conveying information securely. The basic aim of cryptography is to allow the intended recipients of a message to receive the message properly while preventing eavesdroppers from understanding the message. Cryptanalysis is the science and study of method of breaking cryptographic techniques i.e. ciphers. In other words it can be described as the process of searching for flaws or oversights in the design of ciphers [2].
One of the first knapsack ciphers was suggested by Merkle and Hellman in 1978 [3]. It represented one of the initial attempts at a public key cryptosystem. While the cipher is based on an NP_complete problem [4].

[+] **Received on 8/10/2013 , Accepted on 23/3/2015**
[*] **Assistant Prof./College of Electrical and Electronic Engineering Technologies Middle Technical University**
[**] **Master Student**

Kunikatsu Kobayashi etc have been proposed a knapsack cryptosystem based on three knapsacks. The secret sequence corresponding two knapsacks are non-superincreasing, while the other one is superincreasing. This knapsack cryptosystem uses nonlinear multiplicative operation in the encryption, and therefore is thought to be secure against the low density attack using LLL algorithm. On the other hand, the two secret sequences are non-superincreasing in this knapsack cryptosystem, and therefore it is thought to be secure against the attack using Shamir algorithm.

This paper focuses on attack on knapsack cryptosystem based on three knapsacks using Genetic Algorithm (GA) [1]. Genetic Algorithms are optimization and search technique based on the principles of genetic and natural selection [5]. They contain three main operators: selection, crossover and mutation [6].

<u>A knapsack based on multiple knapsacks:</u>

A. Key- generation:

Each entity creates a public key and a corresponding private key.

1. An integer n is fixed as a common s system parameter.
2. Alice should perform steps 3 – 7.
3. Choosing the secret keys $A=(a_1, a_2, \ldots, a_n), B=(b_1, b_2, \ldots, b_n)$, and $E=(e_1, e_2, \ldots, e_n)$ corresponding to the three knapsacks, where A and B are positive integers and E are non-negative integers. The secret sequences A, B and E are chosen so as to satisfy Conditions 1 and 2 below.

Condition 1.

The secret sequences $A=(a_1, a_2, \ldots, a_n), B=(b_1, b_2, \ldots, b_n)$, are non-superincreasing, i.e.,

$$a_k \le \sum_{i=1}^{k-1} a_i \quad \& \quad b_k \le \sum_{i=1}^{k-1} b_i \qquad (1)$$

for every $k = 2,\ldots,n.$

Condition 2.

$$a_k b_k - \left(\sum_{i=1}^{k-1} a_i\right)\left(\sum_{i=1}^{k-1} b_i\right) + e_k - \sum_{i=1}^{k-1} e_i + a_1\left(2^{n-1} - 2^{k-1}\right)\left(a_k - \sum_{i=1}^{k-1} a_i\right) + b_1\left(2^{n-1} - 2^{k-1}\right)\left(b_k - \sum_{i=1}^{k-1} b_i\right) > 0 \qquad (2)$$

For every $k = 2, \ldots, n.$

For each $k = 2, \ldots, n$, assume that the components $a_1, a_2, \ldots, a_{k-1}; b_1, b_2, \ldots, b_{k-1}; e_1, e_2, \ldots, e_{k-1}$; chosen so far. Then the next components a_k, b_k and e_k are chosen easily so as to satisfy (1) and (2) simultaneously. This is because the inequality (2) is equivalent to the inequality

$$e_k > -a_k b_k + \left(\sum_{i=1}^{k-1} a_i\right)\left(\sum_{i=1}^{k-1} b_i\right) - \sum_{i=1}^{k-1} e_i$$
$$- a_1\left(2^{n-1} - 2^{k-1}\right)\left(a_k - \sum_{i=1}^{k-1} a_i\right) \quad \text{.....(3)}$$
$$- b_1\left(2^{n-1} - 2^{k-1}\right)\left(b_k - \sum_{i=1}^{k-1} b_i\right)$$

and therefore one can choose e_k to satisfy (2) after choosing a_k, b_k to satisfy (1).

The secret sequences A and B are superincreasing by Condition 1. On the other hand, since

$$a_k b_k \le \left(\sum_{i=1}^{k-1} a_i\right)\left(\sum_{i=1}^{k-1} b_i\right)$$

by Condition 1, it follows from Conditions 1 and 2 that the secret sequences E is superincreasing, i.e.,

$$e_k > \sum_{i=1}^{k-1} e_i$$

holds for every $k = 2, \ldots, n$.

4. Selecting a random integer

$$p > \left(\sum_{i=1}^{n} a_i\right)\left(\sum_{i=1}^{n} b_i\right) + \sum_{i=1}^{n} e_i$$

5. Select a random integers u, v, such that $\gcd(u, p) = 1$, $\gcd(v, p) = 1$.
6. Find inverse of $w \bmod u$
7. Based on the secret key (A, B, E, p, u, v), calculate the public keys
   $F = (f_1, f_2, \ldots, f_n)$, $G = (g_1, g_2, \ldots, g_n)$, and $H = (h_1, h_2, \ldots, h_n)$ by
   $f_i = ua_i \pmod p$,
   $g_i = ub_i \pmod p$,
   $h_i = uve_i \pmod p$
   For each $i = 1,2,..., n$
8. Alice's public key is F, G, H; Alice's private key is (A, B, E, p, u, v).

For the above example:

1. n=8.
2. Alice should perform steps 3 – 7.
3. Choosing asuperincreasingsequence $a = (3,3,6,12,24,48,96,192)$,
   $b = (4,3,7,14,28,56,112,224), e = (2,6,9,19,38,75,152,303)$

4. Selecting $p = 172641$.
5. Selecting $u = 121, v = 109$.
6. F = (363 , 363 , 726 , 1452, 2904 , 5808, 11616 , 23232),
   G = (436, 327, 763, 1526, 3052, 6104, 12208, 24416),
   H = (26378, 79134, 118701, 77950, 155900 , 125970,
   105677, 25524) .
7. Alice's public key is F, G, H ; Alice's private key is (A, B, E, p, u, v) .

B.encryption and decryption:

Bob encrypts a message m for Alice, which Alice decrypts.
1. *Encryption.* Bob should do the following:

(a) Obtain Alice's authentic public key F, G, H .

(b) Represent the message m as a binary string of length $n, m = m_1 m_2 ... m_n$.

(c) Compute the integer $c = c_1 c_2 + c_3$,

Where
$c_1 = f_1 m_1 + f_2 m_2 + ... + f_n m_n$.

$c_2 = g_1 m_1 + g_2 m_2 + ... + g_n m_n$.

$c3 = h_1 m_1 + h_2 m_2 + ... + h_n m_n$.

(d) Send the ciphertext c to Alice.

2. *Decryption.* To recover plaintext m from c , Alice should do the following:

(a) Compute $D(C) = u^{-1} v^{-1} c (\text{mod } p)$.

Let u^{-1} and v^{-1} be the modular inverses of u (mod p) and v (modp), respectively.

(b) Find the message bits are $m_i = m_1, ..., m_n, m_i \in \{0,1\}$, such that

$$D(C) = \left(\sum_{i=1}^{n} a_i m_i \right) \left(\sum_{i=1}^{n} b_i m_i \right) + \sum_{i=1}^{n} e_i m_i .$$

The plaintext $m_i = m_1, ..., m_n$ is calculated based on the algorithm below.
Input: a positive integer n, a non-negative integer $D(C)$ and a triplet (A,B,E) of sequences of integers satisfying Conditions 1 and 2
Output: A sequence m of 0 and 1 of length n
1. for k = n down to 1 do
2. Set

$$d = \left(\sum_{i=k+1}^{n} a_i m_i + a_k \right) \left(\sum_{i=k+1}^{n} b_i m_i + b_k \right) + \sum_{i=k+1}^{n} e_i m_i + e_k$$

3. If $D(C) \geq d$ then $m_k = 1$ else $m_k = 0$.
4. Set $m = m_1, ..., m_n$.

Note here that each of the sums $\sum_{i=k+1}^{n} a_i m_i + a_k$, $\sum_{i=k+1}^{n} b_i m_i + b_k$ and $\sum_{i=k+1}^{n} e_i m_i + e_k$ in the algorithm is interpreted as 0 in the case of k = n.

For the above example:

Bob encrypts amessage $m = R$ for Alice, which Alice decrypts.
1. *Encryption* .Bob should do thefollowing:

(a) Obtain Alice's authentic public key

F = (363, 363, 726, 1452, 2904, 5808, 11616, 23232),
G = (436, 327, 763, 1526, 3052, 6104, 12208, 24416),
H = (26378, 79134, 118701, 77950, 155900 , 125970, 105677, 25524) .

(b) $m = R = 01010010$.

(c) $c1 = 13431$.
$c2 = 14061$.
$c3 = 262761$.
$c = c_1 c_2 + c_3, = 189116052$

(d) Send the ciphertext c to Alice.

2. *Decryption* .To recover plaintext m from c :

(a) $dc = 14528$.

(b) The message bits are $m = (01010010) = R$.

Genetic algorithms:

Genetic algorithms were developed by John Holland as a modification of what is called evolutionary programming [4].
Holland's idea was to construct a search algorithm modeled on the concepts of natural selection in the biological sciences. The result is a directedrandom search procedure. The process begins by constructing a random population of possible solutions. This population is used to create a new generation of possible solutions which is then used to create another generation of solutions, and so on. The best elements of the current generation are used to

create the next generation. It is hoped that the new generation will contain "better" solutions than the previous generation [4,9].
Three processes which have a parallel in human genetics are used to make the transition from one population generation to the next. They are selection, mating and mutation. The basic genetic algorithm cycle based on these three processes is shown in the figure 1. Selection process determines which strings in the current generation will be used to create the next generation. The mating process determines the actual form of the strings in the next generation. At this point, two of selected parents are paired. The final step is one of mutation. A fixed small mutation probability is set at the start of the algorithm. Bits in all the new strings are then subject to change based on this mutation probability [4].

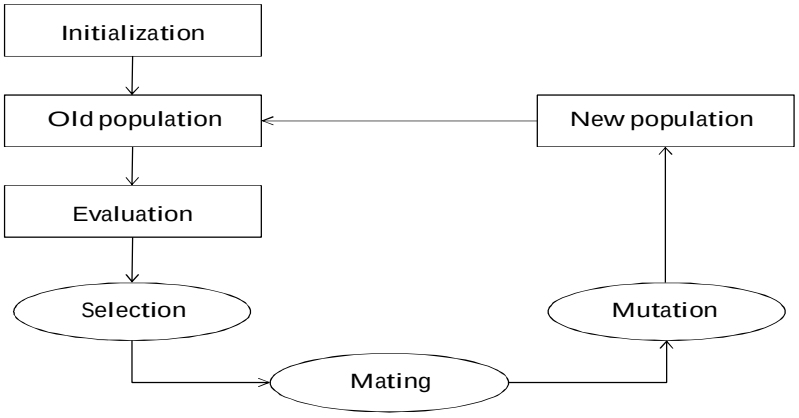


Figure (1):The basic genetic algorithm cycle.

Cryptanalytic on knapsack cipher using genetic algorithm:

Spillman[4] suggested genetic algorithm to solve the knapsack problem. Fig. 2 shows the Markle-Hellman cryptosystem and cryptanalysis by means of genetic algorithm. The cryptanalysis starts from cipher text, which has an integer form. Each number represents a target sum of hard knapsack problem. The goal of the genetic algorithm is to translate each number into the correct knapsack, which represents the ASCII code for the plaintext characters.

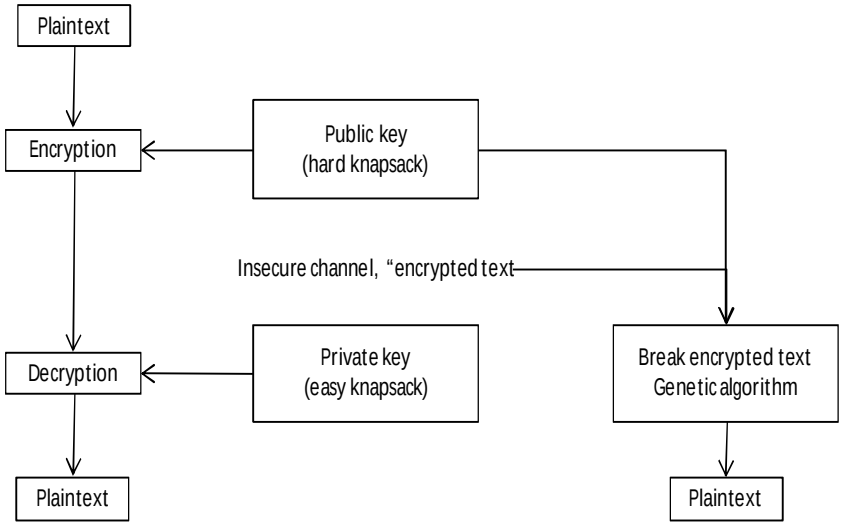


Figure (2): Markle-Hellman cryptosystem and cryptanalysis by means of Genetic Algorithm.

Encoding:

The following restriction have been made for encoding

(1) Only the ASCII code will be encrypted.

(2) The superincreasing sequence will have 8 elements; this number of elements guarantees that each character has a unique encoding.

Initialization:

A random population of chromosomes (binary string 0's and 1's) is generated. The number of bits in each chromosome is equal to the number of elements key (i.e. 8).

Evaluation:

In our work we will use the following fitness function [4] to evaluate the generated individuals. Based on the fitness value obtained, it can be determined whether the optimal solution is reached or not.

$$fitness = \begin{cases} 1-\left(\frac{|T \text{ arg } et - Sum|}{T \text{ arg } et}\right)^{\wedge}\frac{1}{2} & if \;\; Sum \leq T \text{ arg } et \\ 1-\left(\frac{|T \text{ arg } et - Sum|}{MaxDiff}\right)^{\wedge}\frac{1}{6} & if \;\; others \end{cases}$$

Where,

MaxDifference = max (Target, FullSum − Target)

Target is the ciphertext.

x is the chromosome.

$\text{FullSum} = x * F'$

$$\text{sum} = \left(x * F'\right) * \left(x * G'\right) + \left(x * H'\right).$$

Based on the fitness function given in the equation above the fitness value evaluates how the given sum is close to the target value for the knapsack. Fitness value 1 indicates an exact match with the target sum for the knapsack. If the value of sum is greater then targets then it has a lower fitness value of chromosome, in this way it produces the infeasible solution. If the value of sum is less then target then it will produce a high fitness value and produce feasible solutions. Feasible solutions have a greater chance of being followed by the algorithm. Small differences between the current chromosome and the target sum should be amplified.

Selection:

The important part of algorithm is selection of a new population. Selection of individuals is done according to their fitness value obtained. In our work we used the stochastic universal sampling selection. Stochastic universal sampling selection procedure may be implemented as follows:

1. The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means dividing the fitness value of each individual by the sum of all fitness values, so that the sum of all resulting fitness values equals 1.
2. The population is sorted by descending fitness values.
3. Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of the last individual should be 1 (otherwise something went wrong in the normalization step).
4. A random number *R* between 0 and 1 is chosen.
5. The selected individual is the first one whose accumulated normalized value is greater than *R*.

Elite:

Elite children are the individuals in the current generation with the best fitness values. These individuals automatically survive to the next generation [10].

Crossover:

The single point crossover operation applied in the algorithm.

Mutation:

After crossover is performed, mutation takes place. Bit inversion is the type of mutation is used in this work.

Results:

Genetic algorithm has been applied to cryptanalysis of knapsack cipher successfully in short time.

This paper used Genetic Algorithms to cryptanalysis knapsack cipher based on three knapsacks. The knapsack cipher is with a knapsack sequence of size 8.

The number of generation is changed, the population size is changed, the selection type is stochastic universal sampling, the crossover type is single point crossover and this point is selected randomly, the crossover probability is 0.5, the mutation type is reversing, the mutation probability is 0.3, three bits from eight bits are reversed in the mutation and elite is 0.2 are used in this paper.

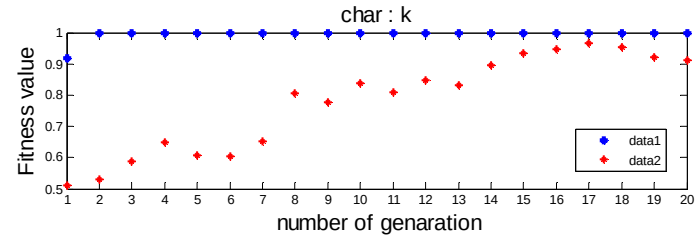
The private key used for the knapsack cryptosystem was $a = (3,3,6,12,24,48,96,192)$,

$b = (4,3,7,14,28,56,112,224), e = (2,6,9,19,38,75,152,303)$

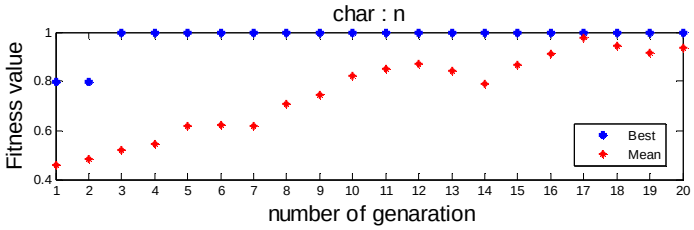
and values of u, w are randomly selected at each run. The value of w^{-1} is computed according to the values u, w. The public key for the knapsack cryptosystem is changed at each run according to the value u.

In our work the word (knapsack) is encrypted. The total time to obtain the word macro is 0.04944second. The software (MATLAB 2009A) has been used. Pentium CORE^{TM}i5, processor 2.30GHZ, and RAM 4GB. Figure (3) shows the Number of generation versus Population size .When the population size increases the number of generation decreases. Figure (3) shows the best and mean fitness for each character in the word macro. The best fitness of character k becomes 1 in the generation 2 therefore the character k is obtained in the generation 2 as shown in Figure (3-a). While the character n is obtained in the generation 3

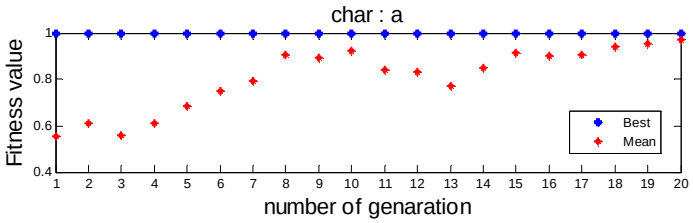
shown in Figure (3-b) and so on for the others. Figure (4) shows the best fitness of all character in the word knapsack. The character a is obtained first, then the character k and s, then the character n, then the characters c, then the character k, then the character a, then the character p.



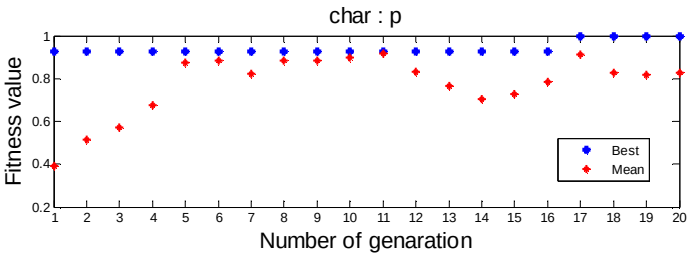
-a-



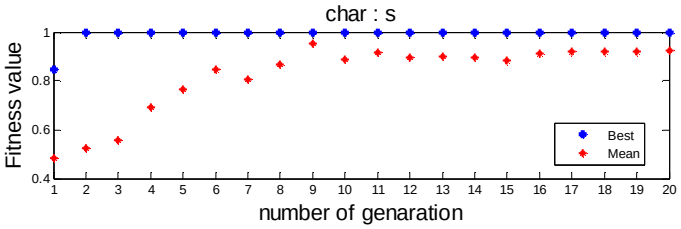
-b-



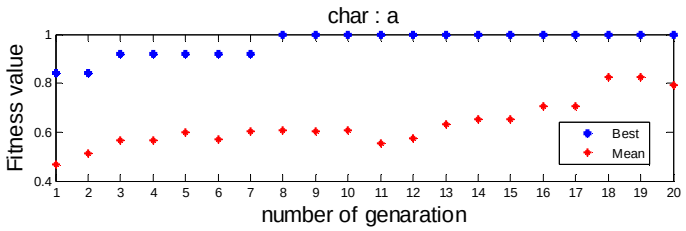
-c-



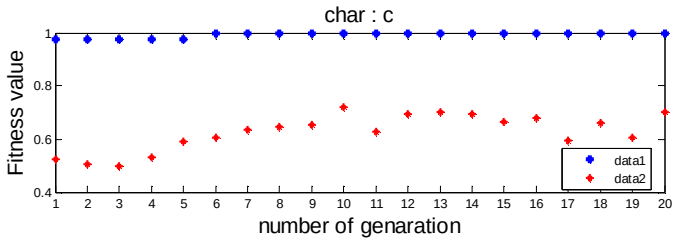
-d-



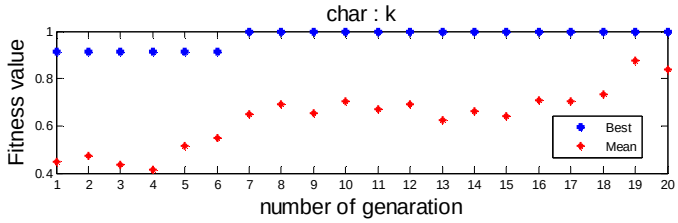
-e-



-f-



-g-



-h-

Figure (3): Best and mean fitness for each character in the word knapsack

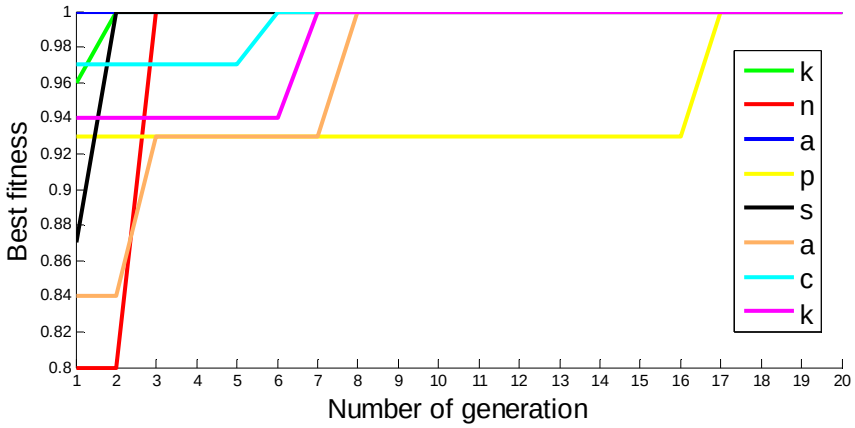


Figure (4): Best fitness versus Number of generation

Conclusion:

The knapsack cipher based on three knapsacks is thought to be secure both against the low density attack using LLL algorithm and against the attack for computing secret keys from public keys using Shamir algorithm.

This paper showed that the genetic algorithm can be solve this type of knapsack successfully.

This paper indicates that the efficiency of genetic algorithm attack on knapsack cipher can be improved by variation of mutation, crossover operation and size of population. The results are

worse when the size of population decreases. The initial population size is inversely proportional to number of generations.
The genetic algorithm offers a powerful tool for the cryptanalysis of knapsack cipher.

References:

1. R. Geetha, and L. Balasubramanian, "Genetic *Algorithm solution for Cryptanalysis of Knapsack Cipher with Knapsack Sequence of Size 16* ,"*International Journal of Computer Applications* ,11(35):(0975 – 8887), *December* 2011.
2. G. Poonam, and S. Aditya, "*An Improved Cryptanalytic Attack on Knapsack Cipher using Genetic Algorithm*," *International Journal of Information Technology*, 3(3):(145-15), 2006.
3. R. C. Merkle and M. E. Hellman, "Hiding Information and Signatures in Trapdoor knapsacks,". IEEE Transactions on Information Theory.IT-24: 525-530, 1978.
4. R. Spillman, "*Cryptanalysis of knapsack ciphers using genetic algorithms*," C*ryptologia*, 17(4):367–377, October 1993.
5. R. L. Haupt, and S. Ellen Haupt, " Practical *genetic algorithms*," 3rd addition, Wiley interscience, 2004.
6. A. P. Engelbrecht, "Computational *Intelligence An Introduction*," 2nd, 2007.
7. G. Poonam, S. Aditya, and D.C. Agarwal, " *An Enhanced Cryptanalytic Attack on Knapsack Cipher using Genetic Algorithm*," proceedings of world academy of science, engineering and technology, 12:(1307-688), march 2006.
8. J. Alfred, "*Handbook of Applied Cryptography*," CRC, 1996.
9. R. Spillman, "*Solving Large Knapsack Problems with a Genetic Algorithm*," *IEEE*, (632-637), 1995.
10. *"Genetic Algorithms and Direct Search Toolbox™ 2"* User's Guide, MATLAB. 2009.