Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp203-206)

An Effective Repair Procedure based on Quantum-inspired Evolutionary Algorithm for 0/1 Knapsack Problems

Zhongyu Zhao, Xiyuan Peng, Yu Peng
Department of Automation Measurement and Control
Harbin Institute of Technology, Harbin, 150001, China P.O.Box 339, China.

Enzhe Yu*
Electronic Telecom. IS Team, Samsung Data System,
Seoul 100-742 Korea.

Abstract: A new repair method based on QEA for 0/1 knapsack problems is proposed. In this approach, the qubit chromosome is used as heuristic knowledge to evaluate each element for the knapsack. The main idea is to delete the knapsack elements in the ascending order of qubit chromosome's probability value whilst avoid violating the constraints on its capacity. To minimize the influence of initialization, three different methods are adopted for obtaining the initial probability. Experimental results showed that the proposed method is promising.

Key-Words: Quantum-inspired evolutionary algorithm, repair algorithm, 0/1 knapsack problems

1. Introduction

Quantum-inspired evolutionary algorithm (QEA) is recently proposed by Han, which is a probabilistic model evolutionary algorithm based on the concept and principles of quantum computing[1-3].It can treat the balance between exploration and exploitation more easily compared to conventional GAs. When QEAs are applied to knapsack problems, unfeasible solutions are often generated by observation operator. That is, generated solution does not always satisfy constraint conditions (a solution that exceeds the capacity of the knapsack). The random repair procedure is used in [1-3] for deriving feasible solutions from unfeasible ones. In this algorithm, elements are selected in random order and no heuristic knowledge is used, therefore the algorithm cannot effective exploit the search space.

QEA uses a novel representation that is based on the concept of qubits. The qubit chromosome represents the probability of obtaining different state. In this paper, we propose a new repair procedure based on the qubit chromosome. In this approach, we use qubit chromosome as heuristic knowledge to evaluate each element for the knapsack. So we select elements for deletion in the knapsack in the increasing probability value order.

The rest of this paper is structured in the following way. Section 2 shows conventional constraint handling techniques for 0/1 knapsack problems. In section 3 we describe the proposed repair procedure. The next section presents experimental evaluation. Finally, section 5 concludes this paper.

2. Conventional Constraint Handling Techniques

There are constraint handling techniques used in evolutionary computation as follows: various penalty function, repair algorithms, specialized representation and hybrid algorithms, etc[4]. Of the known techniques, penalty function and repair methods are popularly used for 0/1 knapsack problems.

2.1 Penalty Methods

In most applications to constrained optimization problems, the penalty function method has been used, because each one of them can be easily applied to any problem without much change in the algorithm[5]. The main idea of penalty functions is to penalize the evaluation of infeasible solutions so that it is not competitive with the evaluations of feasible solutions.

This evolution is done in the following manner:

* The corresponding author. Email: enzhe@snu.ac.kr

$$f(x) = \begin{cases} \sum_{i=1}^{n} p_i x_i & \text{if } \sum_{i=1}^{n} w_i x_i \le c \\ \sum_{i=1}^{n} p_i x_i - \text{pen}(x) & \text{if } \sum_{i=1}^{n} w_i x_i > c \end{cases} \quad (1)$$

where $x = (x_1 ... x_m)$, x_i is 0 or 1, p_i is the profit of item i, w_i is the weight of item i, and C is the capacity of the knapsack.

There are many possible strategies for assigning the penalty function. Here, only two cases were considered, *i.e.* the growth of the penalty function is linear penalty and quadratic penalty with respect to the degree of violation, respectively:

$$A_p[1]: pen_1(x) = \rho(\sum_{i=1}^{n} w_i x_i - C), \quad (2)$$

$$A_p[2]: pen_2(x) = (\rho(\sum_{i=1}^{n} w_i x_i - C))^2, \quad (3)$$

where ρ is $\max_{i=1...m\{p_i / w_i\}}$.

2.2 Repair Methods

Repairing methods are popular in the evolutionary computing community for many combinatorial optimization problems (e.g. the traveling salesman problem, the knapsack problem, and the set covering problem). The process of repairing infeasible individuals is related to learning, which is local search in general or local search for the closest feasible solution in particular[5].

In repair methods, the profit $f(x)$ of each string is determined as:

$$f(x) = \sum_{i=1}^{n} p_i x_i', \quad (4)$$

where x' is a repaired individual which is selected from the original vector x.

There are two repair methods which follow the outline of this repair procedure. They may differ in selection procedure, which chooses an element for removal from the knapsack.

$A_r[1]$ (random repair): selects a random element from the knapsack, which is used in[2].

$A_r[2]$ (first fit algorithm): selects the first available element from the left (right) of the list, which is used in [6]for initialize the population of a GA.

In the random repair and first fit algorithm, no effective heuristic knowledge is used in the local search.

3. The Proposed Repair Method

In QEA, the population of individuals is represented by the qubit chromosome, a vector of probabilities:

$$P_l(x) = (p_l(x_1), ..., p_l(x_i), ..., p_l(x_n)) \quad (5)$$

where $p_l(x_i)$ refers to the probability of obtaining a value of 1 in the i th component of D_l, the population of individuals in the l th generation. $P_l(x)$ is update by quantum gate operation in each generation.

The $p_l(x_i)$ may be used as heuristic knowledge to evaluate the importance of each element for the knapsack. The elements are selected in an ascending order by the chromosome's probability value in order to avoid violating the constraints on knapsack's capacity. The proposed repair algorithm is shown in Figure 1.

Procedure repair (x)
Begin
knapsack-overfilled = false
$x' = x$
If $\sum_{i=1}^{n} w_i x_i > c$
then knapsack-overfilled := true
while (knapsack-overfilled) *do*
begin
i := select an element in the increasing order of the probability value to remove
i.e., x'[i] := 0
if $\sum_{i=1}^{n} w_i x_i \le c$
then knapsack-overfilled := false
end
end

Fig.1 First probability value algorithm ($A_r[3]$)

4. Experiments

4.1 Test Problems

In this section we carry out some experiments with different number of objects (n=100, 200 and 500). For each experiment we use strongly correlated sets of data as follow:

$$w_i = \text{uniformly random}[1,10), \quad (6)$$

$$p_i = w_i + 5, \quad (7)$$

$$C = 1/2 \sum_{i=1}^{n} w_i \quad (8)$$

where w_i is the weight of item i, p_i is the profit of item i, and C is the capacity of the knapsack.

4.2 Initialization
In order to minimize the influence of initialization, three different methods are used to obtain the initial probability.

4.2.1 Uniform
Each item is selected with equal probability, independent of the remaining elements. In order to obtain an expected number of selected elements of $nc/\sum_{i=1}^{n} w_i$, each element is selected with a probability equal to $c/\sum_{i=1}^{n} w_i$.

$$(p_0(x_1),..., p_0(x_n)) = (c/\sum_{i=1}^{n} w_i ,..., c/\sum_{i=1}^{n} w_i). \quad (9)$$

4.2.2 Proportional
Each element is selected with a probability proportional to its ration between profit and weight, which can be expressed as following:

$$(p_0(x_1),..., p_0(x_n)) \propto (p_1/w_1 ,..., p_n/w_n). \quad (10)$$

4.2.3 Probabilistic Seed
In this method, starting form a feasible solution, an initial probability value is obtained:

$$(p_0(x_1),..., p_0(x_i),..., p_0(x_n))$$

Where for all i=1,…,n:

$$p_0(x_i) = \begin{cases} \alpha \text{ if item } x_i \text{ is selected} \\ 1-\alpha \text{ if item } x_i \text{ is not selected} \end{cases} \quad (11)$$

In this paper, we fix the α value to 0.95.

4.3 QEA for 0/1 Knapsack Problems
Fig.2 shows the procedure of QEA. The procedure of QEA is explained as follows:

i) $p_0(x_i)$ is initialized as described in section 3.2.

ii) This step generates binary solution X_t by observing the states of $p_0(x)$. One binary solution is formed by selecting each bit using the probability of qubit.

iii) Binary solution X_0 is evaluated and repaired. The initial best solution is into B, where $B_0 = \{b_0(x_1), b_0(x_2), \Lambda, b_0(x_n)\}$, and $b_0(x_i)$ is the same as $x_0(i)$ at the initial generation.

iv) Terminal condition is defined as following:

$$C_t = \frac{1}{n}\sum_{i=1}^{n} \left|1 - 2 * |p_t(x_i)|^2\right| \text{ f } \lambda \quad (12)$$

where λ value is fixed to 0.96 in this paper.

v, vi) In the while loop, binary solution in X_t is formed by observing the states of P_{t-1} as in step ii), and each binary solution is evaluated and repaired for the fitness value.

Procedure of QEA

begin
t=0
Initialize p_t
Make $x(t)$ by observing the states of p_t
Evaluate and repair x_t, save the best solutions among x_t into B_t
while (*not* terminal condition) *do*
begin
$t = t + 1$
Make x_t by observing the states of p_{t-1}
Repair and evaluate x_t
Update p_t using H_ε -gates
Store the best solutions among B_{t-1} and x_t into B_t
end
end

Fig.2 The procedure of QEA

vii) In this step, Q-bit individuals in P_t are updated by applying H_ε-gates as used in [3]. The following rotation gate is used as a H_ε-gate in QEA:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \quad (13)$$

where $\Delta\theta_i$, i=1,2,…,m, is a rotation angle of each Q-bit. The rotation angle $\Delta\theta_i$ is showed in Table 1.

Table 1 Lookup table of $\Delta\theta_i$, where f(·) is the fitness function; x_i and b_i are the i-th bits of the binary solution x and the best solution b

	$x(i)$	$b(i)$	$\Delta\theta_i$
$f(x)$ p $f(b)$	0	1	-0.01 π
	1	0	0.01 π
other			0

viii) The best solutions among B_t and P_{t-1} are selected and stored into B_t, and if the best solution stored in B_t is a better solution fitting than the stored best solution b, the stored solution b is replaced by the new one. Until the termination condition is satisfied, QEA is running in the while loop.

4.4 Experimental Results and Discussions
Table 2 to 4 show the average profits and iterations from 20 independent experiments for the 100, 200

and 500 objects problems, respectively. As shown in the tables, five ways are considered for the evaluations of the individuals in combination with three initializations.

Table 2 Experimental results of the knapsack problem of n=100, t represents iterations

	uniform		proportional		probabilistic seed	
	profit	t	profit	t	profit	t
$A_p[1]$	569.1	334	552.8	378	557.7	298
$A_p[2]$	563.2	301	539.0	324	556.5	303
$A_r[1]$	581.2	323	572.6	397	574.3	346
$A_r[2]$	580.9	349	574.1	468	573.7	382
$A_r[3]$	586.1	1431	591.3	2716	580.5	1493

Table 3 Experimental results of the knapsack problem of n=200, t represents iterations

	uniform		proportional		probabilistic seed	
	profit	t	profit	t	profit	t
$A_p[1]$	1109.5	382	1050.4	370	1095.0	338
$A_p[2]$	1100.1	350	1040.6	426	1093.2	351
$A_r[1]$	1143.4	404	1111.0	486	1131.9	420
$A_r[2]$	1138.9	427	1135.3	680	1129.5	468
$A_r[3]$	1148.3	691	1170.0	683	1132.0	651

Table 4 Experimental results of the knapsack problem of n=500, t represents iterations

	uniform		proportional		probabilistic seed	
	profit	t	profit	t	profit	t
$A_p[1]$	2725.0	383	2608.2	452	2694.1	333
$A_p[2]$	2711.9	345	2583.5	308	2687.7	346
$A_r[1]$	2787.3	466	2739.4	692	2767.6	474
$A_r[2]$	2783.4	456	2785.4	907	2772.3	502
$A_r[3]$	2800.8	569	2828.8	1153	2771.0	538

Comparing different initialization, QEAs designed by proportional initialization yielded superior results as compared to all others. QEAs with uniform initialization outperformed the probabilistic seed initialization. This shows the performance of QEA for 0/1 knapsack problem strongly depends on the choice of an initialization method.

Comparing different repair procedure, roughly speaking QEAs using first probability value repair algorithms have the best profit and the most iteration. The only exception is under the conditional of probabilistic seed initialization for 500 objects, because the chromosome's probability value is set by a feasible solution and the probability value can't completely reflect the importation of every chromosome. QEAs using penalty function are worst due to their weak ability to exploit the mathematical structure of the constraint.

5. Conclusions

In this paper, we propose a new repair procedure for QEA to solve the 0/1 knapsack problems. Experimental results demonstrate this proposed repair algorithm can effective exploit the search space and improve the performance of QEA for 0/1 knapsack problems.

References

[1] Han KH, Kim JH, Genetic quantum algorithm and its application to combinatorial optimization problem, *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, Shanghai, China, 2000 pp4066.

[2] Han KH, Kim JH, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Transactions on Evolutionary Computation*, Vol.6, No.6,2002 pp580.

[3] Han KH, Kim JH, Quantum-inspired evolutionary algorithms with a new termination criterion, H gate, and two-phase scheme, *IEEE Transactions on Evolutionary Computation*, Vol.8, No.2 2004 pp156

[4] Michalewicz etc, *Evolutionary algorithms for constrained engineering problems*, Pergamon Press Inc, Tarrytown, NY, USA Vol.30, No.4, 1996 pp851

[5] Thomas Back, David Fogel, Zbigniew Michalewicz, *Evolutionary computation vol 2 advanced algorithms and operators*, Bristol and Philadelphia: institute of physics publishing, 2000

[6] Hinterding R, Mapping, order-independent genes and the knapsack problem. *Proceedings of the IEEE Conference on Evolutionary Computation*, *ICEC*, Shanghai, China, 1994 pp13