SOLVING MULTIPLE-CONTAINER PACKING PROBLEMS USING PSEUDO-MEIOSIS GENETIC ALGORITHM

Dr. Bara'a Ali Attea Deldar Ibrahem Abdo Al-Rahman
University of Baghdad
College of Science
Department of Computer Science

ABSTRACT

Knapsack problems are a class of common but difficult (NP-complete or NP – hard) problems. Since, it is believed that no knapsack problem algorithm can be constructed whose computation time optimality increases as any polynomial function of the problem size. There is a variety of knapsack–type problems in which a set of entities, together with their values (profits) and sizes, is given, and it is desired to select one or more disjoint subsets so that the total of the sizes in each subset does not exceed given bounds and the total of the selected values is maximized.Diploid representation and dominance operator are advanced operators that attempt to improve upon the power of traditional genetic algórithms. Pseudo – Meiosis Genetic Algorithm (PsM GA) is one form of genetic algorithms that incorporate diploidy structure and dominance mechanism in their genetic search. The goal of this dissertation is to present the application of PsM GA in one of the promising combinatorial optimization problems – the Knapsack Problem (KP). Results obtained concern two types of KP: the 0/1 KP and the Multiple Container Packing Problem, MCPP. Moreover, several aspects are considered in experiments such as, the algorithm used for evaluation of the individuals (fitness evaluation), the number of items (i.e., search space size), the correlation between the weights and the profits of items, and the capacity of the knapsack.

الخلاصة

ان مشاكل الظهر تصنف من ضمن المشاكل العامة (NP-Complete , & NP-Hard) طالما ان المتوقع بانه لا توجد خوارزمية لمشاكل حقيبة الظهر ممكن بناؤها و التي يكون الوقت الأفضل لاحتسابها يتزايد كدالة Polynamial لحجم المشكلة. هنالك تفاوت في أنواع مشاكل حقيبة الظهر من ناحية مجموعة المدخلات سوية مع قيمتها (الأرباح) و أحجامها التي تكون معطاة بطريقة منتخبة لواحدة او اكثر من المجاميع الجزئية المنفصلة بحيث المجموع الكلي للأحجام في كل مجموعة جزئية لا تتجاوز الأوزان المحددة و المجموع الأعلى للقيمة المنتخبة.يعتبر التمثيل المضاعف و عامل التغلب من العوامل المتقدمة اللذان يحسنان قوة الخوارزمية الجينية التقليدية . خوارزمية الانشطار الكاذب هي نوع من الخوارزميات الجينية التي تدمج تقنية الهيكلية المضاعفة و تقنية التغلب في البحث الجيني.بالاضافة الى ذلك توفر خوارزمية الانشطار الكاذب ميكانيكية لاعادة بناء كروموسومات الفرد للجيل القادم. الهدف من هذا البحث هو توضيح تطبيق هذه الخوارزمية (dGA) و (PsMGA) على مجموعة من مشاكل الامثلية. مشكلة حقيبة الظهر

(The Knapsack Problem [KP]) ولقد تم الحصول على نتائج لنوعين مـــن ال KP هـــي : 0/1 KP و
مشكله تعبئة عدة حاويات (Multiple Container Packing Problem [MCPP] .) حيث تم أخذ بنظر الاعتبار
عدة هيئات لتجارب : تقييم الأفراد وعدد المواد و حجم فضاء العينة و العلاقة بين وزن و قيمة المادة و سعة
حقيبة الظهر .

KEY WORDS

Knapsack problem, MCPP, diploid genetic algorithm, NP-hard, post-meiosis chromosome.

INTRODUCTION

Knapsack problems are a class of common but difficult (NP-complete or NP–hard) problems. Since, it is believed that no knapsack problem algorithm can be constructed whose computation time optimality increases as any polynomial function of the problem size [Gra95] [Hop79].

There is a variety of knapsack–type problems in which a set of entities, together with their values (profits) and sizes, is given, and it is desired to select one or more disjoint subsets so that the total of the sizes i: each subset does not exceed given bounds and the total of the selected values is maximized [Mic99].

The 0/1 knapsack problem can be defined in terms of the following simple analogy. For a given set of weights $w[i]$, profits $v[i]$, and capacity W_{max}, the task is to find a binary vector x: $(x[1], \ldots, x[n])$, such that:

$$\sum_{i=1}^{n} x[i] \cdot w[i] \le W_{max} \qquad .(1)$$

and for which

$$f(x) = \sum_{i=1}^{n} x[i] \cdot v[i] \qquad (2)$$

is maximum.[Mic99][Sim01].

Hence, the well-known single-objective 0/1 knapsack problem (KP) is: given a set of items (n), each with a weight $w[i]$ and a profit $v[i]$, with i=1,..,n. The goal is to determine the number of each item to include in the knapsack so that the total weight is less than some given limit W_{max} and the total profit is as large as possible.

The 0/1 knapsack problem can be seen as the variant of the Multiple Container Packing Problem (*MCPP*) with only one container. The MCPP is a combinatorial optimization problem which involves finding the most remunerative assignment of n items with given weights and values to W_{max} containers such that each item is assigned to one container or remains unassigned, and the total weight of each container does not exceed a given maximum. In detail, it can be formulated as follows:

$$\text{maximize } f = \sum_{i=1}^{C} \sum_{j=1}^{n} v_j x_{ij} \qquad (3)$$

$$\text{subject to} \sum_{i=1}^{C} x_{ij} \le 1, \; j = 1, \ldots, n \qquad .(4)$$

and

$$\sum_{i=1}^{C} w_j x_{ij} \le W_{max}, \quad i = 1, \ldots, C \qquad .(5)$$

$x_{ij} \in \{0,1\}$, i=1,…,W_{max}, j=1, …, n,
with $w_j > 0$, $v_j > 0$, $W_{max} > 0$.

Let w_j be the weight and v_j be the value of item j. The variables searched for are x_{ij} (i =1, ..., C. j = 1, ..., n): If item j is assigned to container i, x_{ij} is set to 1, otherwise to 0. The goal is to maximize the total value of all assigned items (1). The n constraints in (4) ensure that each item is assigned to one container at maximum. According to (5), each of the C containers has a total maximum weight W_{max} which must not be exceeded by the sum of the weights of all items assigned to this container.[Rai98].

Looking at the previous knapsack formulations, one may be inspired to ask "do functions like these ever come up in practice?". The answer is resounding "Yes!" It is enough for the reader to imagine some real applications. For example, in Bin Packing Problem (BPP), the goal is to minimize the number of containers necessary to pack all n items while not violating any weight constraint. The values of items do not play a role. Like the KP, the BPP in its general form is NP–hard. Note that the MCPP can also be seen as a complex combination of the KP and the BPP, since the MCPP can be divided into two strongly depending parts which must be solved simultaneously: (a) Select items for packing, and (b) distribute chosen items over the available containers.

One more related problem, which can be seen as a more general form of the BPP, is known under the term *General Assignment Problem (GAP).* A set of jobs (i.e., items) must be assigned to a set of agents (i.e., containers). Each possible assignment has its individual capacity requirements and costs, and each agent has its individual capacity limits. The goal is to distribute all jobs in a way to pay minimal costs while satisfying all constraints [Rai98].

Additionally, a variety of industrial problems can be reduced to knapsack problems, including cargo loading, stock cutting, project selection, budget control, air baggage handling and many other important sectors of economy. Typical cases being the *steel bar cutting stock* problem. For example, the two – dimensional cutting problem requires cutting a plane rectangle into smaller rectangular pieces of given sizes and values to maximize the sum of the values of the pieces cut. This version of the problem appears in the problem of cutting steel or glass plates into required stock sizes to minimize waste. By taking the value of a piece to be proportional to its area, we can formulate the waste minimization problem as one of maximizing the value of the pieces cut. The problem also appears in cutting wood plates to make furniture and paper board to make boxes [6Ada76][Chr77].

CHARACTERISTIC COMPONENTS OF PsM GA

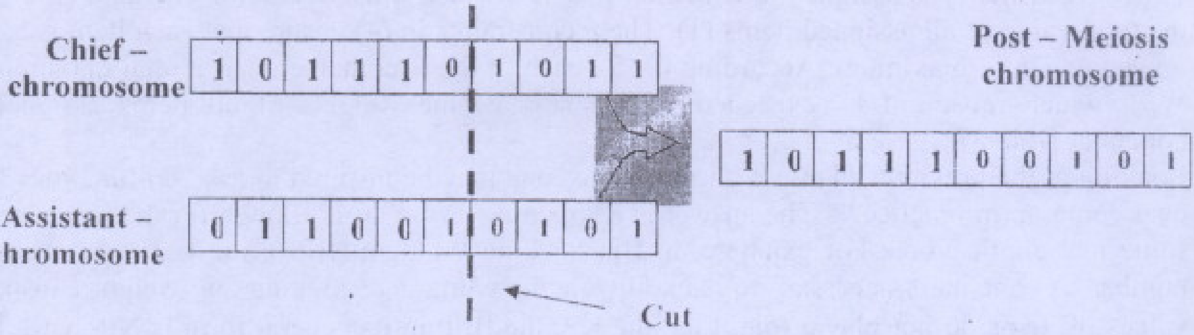
The Pseudo–Meiosis Genetic Algorithm (PsM GA) has a meiosis–like procedure to generate a phenotype from a pair of functionally different chromosomes, unlike a conventional diploid GA with dominance. Additionally, it provides a mechanism for re–pairing the two chromosomes of an individual for the next generations. Below is a detailed explanation of PsM GA.

Individual Representation

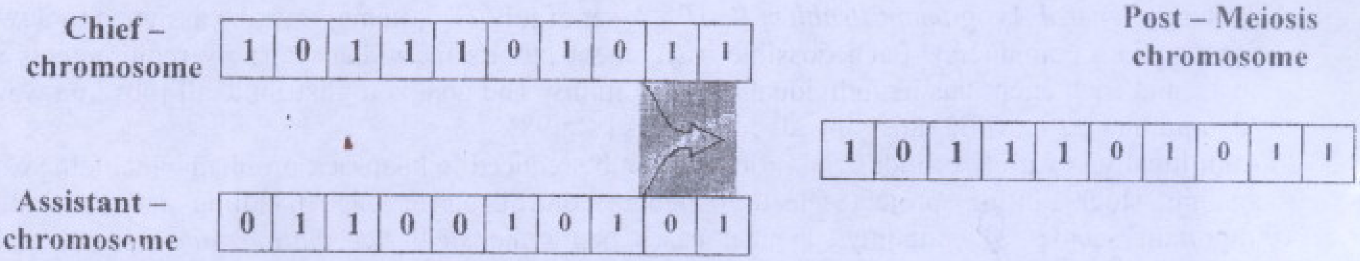
Each individual has two slots to hold a pair of chromosomes; a *chief slot* and an *assistant slot*. The chromosome in a chief slot is called a chief chromosome, and the one in an assistant slot is called an assistant chromosome. Chief and assistant properties do not change during generation. The names, chief and assistant, come from their functions. The chief chromosome is related to the principal search cycle of the GA, whereas the assistant one works only for its chief pair inside the individual [Yos94].

Diploid Stage 1 (Pseudo–Meiosis (PsM) Mapping) and Fitness Evaluation

The Pseudo–Meiosis occurs inside an individual with probability P_{meio} in order to generate a post–meiosis (haploid) chromosome. While for individuals that bypass pseudo–meiosis operation, a copy of the chief chromosome is treated as a post–meiosis one. Consider the following example where Probability $\geq P_{meio}$



The below example is for Probability $< P_{meio}$



Haploid Stage (Selection, Crossover, and Mutation)

This step is very similar to that of ordinary haploid GAs, except that GA operations are applied only to the post–meiosis chromosomes' population. The post–meiosis chromosomes are first evaluated and then selected and reproduced based on their fitness values. The offspring chromosome undergoes crossover, with probability P_c, and mutation, with probability P_m.

Diploid Stage 2(Haploid – to – Diploid Re–Pairing Mechanism)

Each offspring chromosome is pulled back to the chief slot of its parent individual. The chief slot of an individual whose post–meiosis chromosome became extinct because of selection is filled with reproduced, and therefore promising, offspring.

Each assistant chromosome undergoes mutation with probability P_{ma}. Note that, apart from mutation, there is no destructive operation for assistant chromosomes.

The Pseudo–Meiosis GA Layout

The Pseudo–Meiosis mapping, selection, crossover, mutation, and re–pairing form one generation cycle of the PsM GA. The generation (as depicted in the **Fig. (1)**) is repeated until some terminating condition is satisfied.

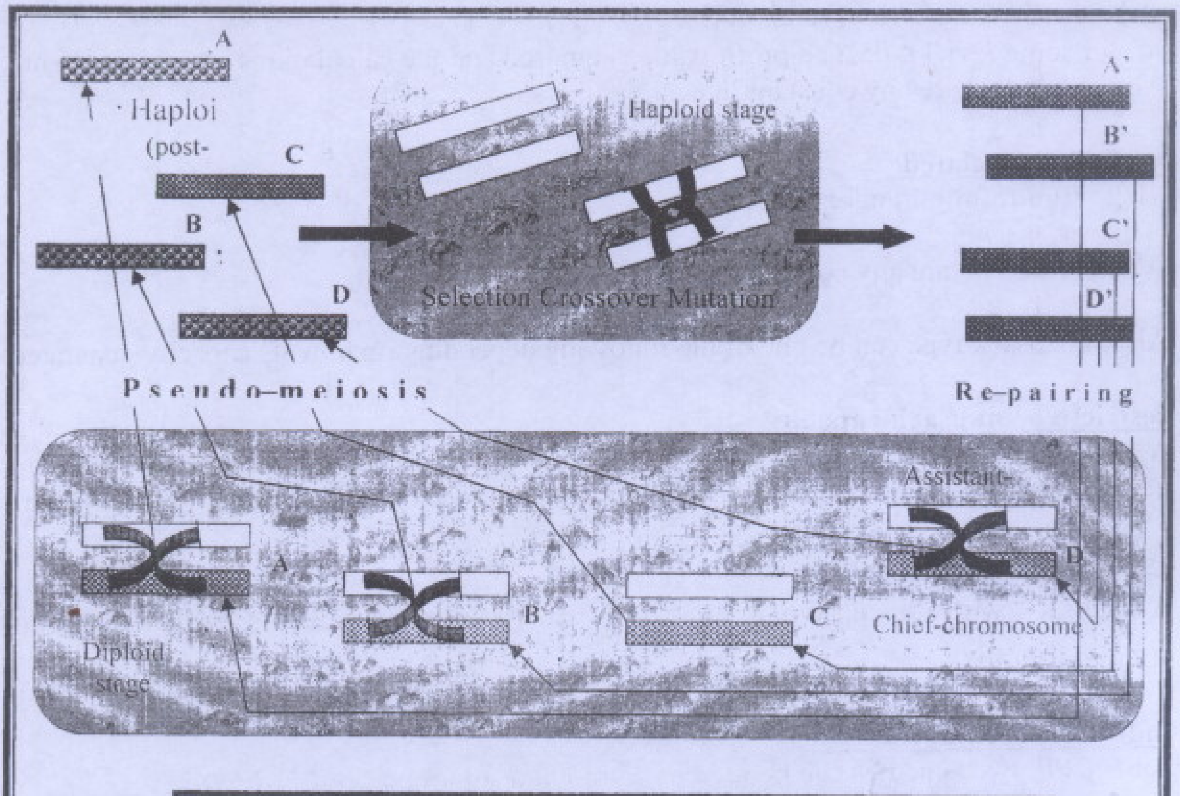


Fig. (1) Genetic operations of the pseudo-Meiosis GA.

Yoshida and Adachi applied their PsM GA to a non–stationary problem Traveling Salesman Problem (TSP) only and they demonstrated that their results were preliminary, and more investigation is needed to use the PsM GA framework for other problems domain.

PsM GA FOR THE MCPP

Following subsections present PsM GA diploid individual representation educate for MCPP, fitness evaluation, PsM GA evolution stages, and finally experimental results.

Problem Intialization

Three randomly generated sets of data are considered that correspond to the correlation between profits and weights of items which in turn affect on the difficulty of the KP problem. [Mic99][Sim01]:

uncorrelated

both vectors $w[i]$ and $v[i]$ are generated at random, using a uniform distribution:
$w[i]$: = (uniformly) random ([1..r]), and
$v[i]$: = (uniformly) random ([1..r]). (6)
Where r is any constant integer number.
Most programming language libraries provide a function to generate normally distributed random numbers.

weakly correlated

vector $w[i]$ is created at random, however, vector $v[i]$ is created with some correlation with $w[i]$.
$w[i]$: = (uniformly) random ([1..r /]), and

$v[i]: = w[i] +$ (uniformly) random ([$-r2, \dots, r2$]). .(7)
(if, for some i, $v[i] \le 0$, such profit value is ignored and the calculations are repeated until $v[i] > 0$). Also, $r1$ and $r2$ are any constant integer numbers.

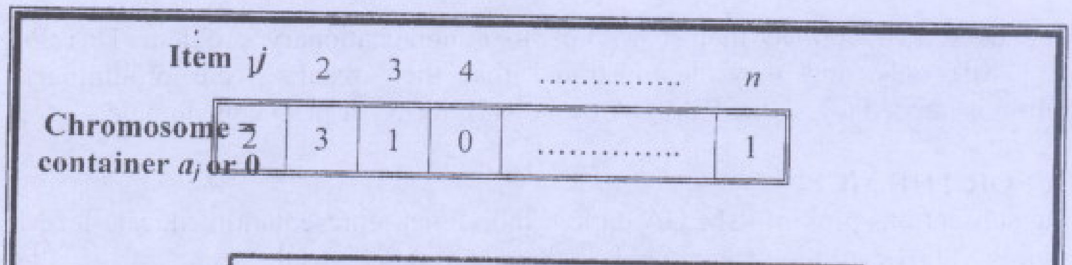
strongly correlated:
$w[i]: =$ (uniformly) random ([1.. υ]), and
$v[i]: = w[i] + r$ . .(8)
Where r and υ are any constant integer number.

Also, knapsack type can be one of the following depending on how its capacity generated:

restrictive knapsack capacity
A knapsack with the capacity of $W_{max} = 2\upsilon$. In this case the optimal solution contains very few items. An area, for which conditions are not fulfilled, occupies almost the whole domain.

average knapsack capacity
A knapsack with the capacity $W_{max} = 0.5\sum_{i=1}^{n} w[i]$. In this case about half of the items are in the optimal solution.

Individual Representation
For MCPP, two schemes can be used to represent a solution vector : [Rai98].
1- A solution is encoded as a vectors of n genes $a(j=1, \dots, n)$. Each gene j represents the number $i(i=1, \dots, C)$ of the container to which item j is supposed to be assigned or the special value 0 if no assignment to any container should be done, see **Fig. (2)**.



Item j	1	2	3	4	....	n
Chromosome = container a_j or 0	2	3	1	0	....	1

.Fig. (2) Direct Encoding Scheme

2- A solution is represented using order based encoding, where the vector a is represented by a permutation of all items $j = 1, \dots, n$ stored in a chromosome $b = (b_1, \dots, b_n)$, see Fig. (3-7).

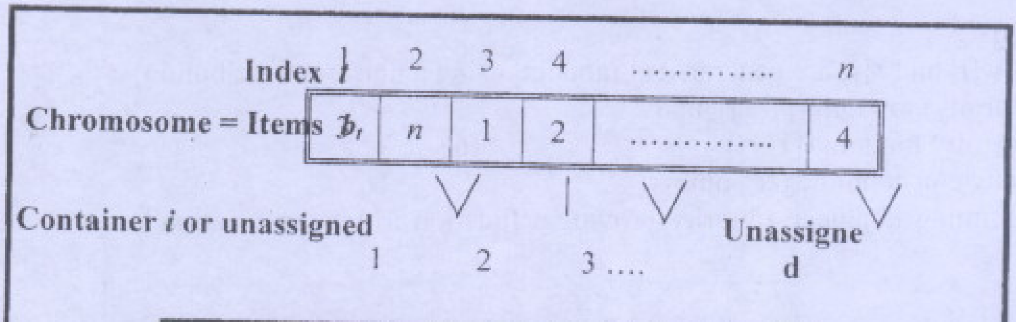


Fig. (3) Order Based Encoding Scheme

For our purpose, we implement the first scheme (i.e., direct encoding scheme) as the solution or chromosome representation with the following consideration:

As an PsM GA individual consists of two slots, a chief slot and an assistant slot, then it can be represented as two vectors of n genes each. One vector for chief slot and the second vector is for assistant slot, see **Fig. (4)** follows.

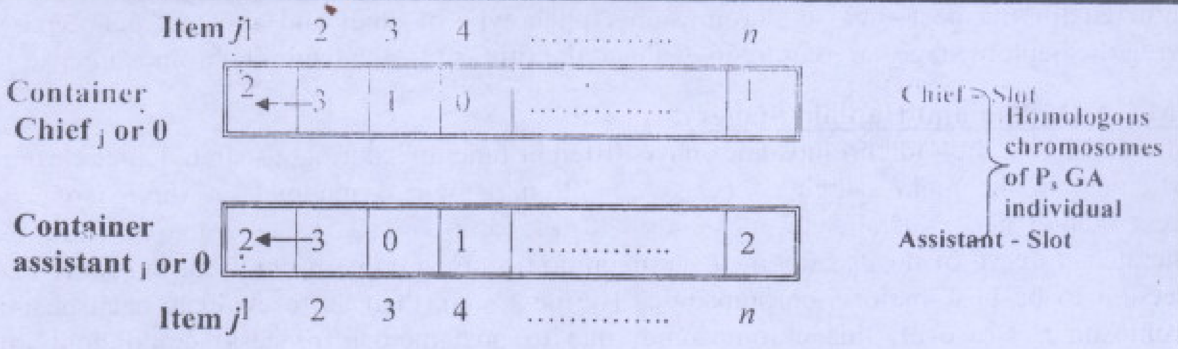

Item j1 2 3 4 .... n
Container Chief $_j$ or 0
2 3 1 0 1
Chief - Slot
Homologous chromosomes of P_s GA individual
Container assistant $_j$ or 0
2 3 0 1 2
Assistant - Slot
Item j^1 2 3 4 .... n

Fig. (4) PsM GA Individual with Direct Encoding Sheme

For the individual example illustrated in the above figure, one can see that chief slot hold a solution in which container number 1 contains items: number 3 and n, container number 2 contains item number 1, container number 3 contains item number 2, while item number 4 dose not exist in any container, and so on. The same explanation can be applied to the assistant slot.

Fitness Evaluation

For the individual representation illustrated in the previous section, the next step is how to evaluate each PsM GA individual (i.e. how to evaluate individual's fitness).

With the direct encoding representation, it is easily possible that solutions are generated which violate constraint concerning the maximum total weight W_{max} of containers. Individual that violate constraint is said to be lethal individual. The algorithm for decoding and possibly repairing a (lethal if any) chromosome a is shown in **Fig. (5)**.

```
Procedure DirectEncoding;
S = 0;
For all item $j$ in random order do
 If $a_j \neq 0$ then
    If $S_{0j}+W_j \leq W_{max}$ then
       $S_{0j} = S_{0j}+W_j$;
       Assign item $j$ to container $a_j$;
    Else
       $a_j = 0$;
done;
```

Fig. (5) Chromosome Decoding with Repair Mechanism Algorithm

First, the current weights of all containers (vector S) are initialized with 0. Then, all items are processed i.ı a random, always different order so that not the same items are favored every time. Each item j is checked if it fits into the container possibly specified in a_j, in which case the item is actually assigned and the current weight of the container S_a, is increased accordingly. If adding item j would result in exceeding the total maximum weight W_{max}, the value of the corresponding gene a_j is set to 0 meaning that the item is not assigned to any container.

Note that the previously mentioned decoding and repairing algorithm is used to evaluate fitness of each PsM GA individual at two different stages. During the diploid stage 1, the fitness function is evaluated for the post–meiosis chromosome (phenotype of chief and assistant genotypes). Later, during the haploid stage, we require to evaluate the fitness of each haploid chromosome.

PsM GA Diploid and Haploid Stages

In the PsM GA, diploid chromosomes have different functions during a search. Chief chromosomes participate in the main search course. When Pseudo–meiosis mapping is applied over the two parent's chief and assistant chromosomes with probability P_{meio}, a post meiosis chromosome is generated. In case of no Pseudo–meiosis mapping occurs, a copy of chief chromosome undergoes selection to be post–meiosis chromosome. The next step, then, is to evaluate each post–meiosis chromosome. However, this chromosome (due to post–meiosis process) could hold infeasible solution. To overcome this problem, the fitness function of each chromosome is evaluated after encoding (and possibly repairing) that chromosome.

After that, a copy of all generated post–meiosis chromosomes are translated to a haploid stage in which the traditional selection (e.g., tournament with elitism), recombination (e.g., one–point crossover) with probability P_c, and mutation with probability p_m are applied.

When these haploid chromosomes are generated, they finally pulled back to their original diploid chromosomes through a haploid – to – diploid repairing mechanism and considered as chief chromosomes for the next generation cycle. However before next generation cycle is started, the assistant chromosome undergoes a mutation with probability P_{ma} contributing, in this way, on producing a variant of its chief chromosome via pseudo–meiosis diploid – to – haploid mapping.

EXPERIMENTAL RESULTS

This section presents experimental results after implementing the PsM GA to solve MCPP. Several instances for the PsM GA and MCPP are considered as illustrated in table 1 and 2.

Table (1)Characteristics of PsM GA Enviroment

	Algorithm Parameter	Setting
1	GA	PsM GA
2	GA – type	Generational – based
3	Selection – type	Tournament with elitism strategy
4	Selection – size	S =2
5	Pseudo – Meiosis type	One – point crossover
6	Pseudo – Meiosis probability	$P_{meio} = 0.3$
7	Haploid Recombination – type	One – point crossover
8	Haploid Recombination – probability	$P_c = 0.6$
9	Haploid mutation – type	Flip – mutation
10	Haploid mutation – probability	$P_m = 0.2$
11	Assistant mutation probability	$P_{ma} = 0.1$
12	Population size	100
13	Stopping – criteria	Max – no. of generation = 150
14	Runs – per experiment	10

Table (2) Characteristics of PsM GA and MCPP Enviroment

	Chromosome length (no. of items)	No. of Container	Container Max – weight	Correlation Weight – Profit of item
PsM GA1	10	3	100	Uncorrelated
PsM GA2	10	3	100	Weakly (r =[-0.8..-1.2,8.8..1.2] , v =[5..95])
PsM GA3	10	3	100	Strongly (r =[0.8..1.2] , v =[5..95])
PsM GA4	20	6	100	Uncorrelated
PsM GA5	20	9	100	Weakly (r =[-0.8..-1.2,8.8..1.2] , v =[5..95])
PsM GA6	10	1	100	Strongly (r =[0.8..1.2] , v =[5..95])

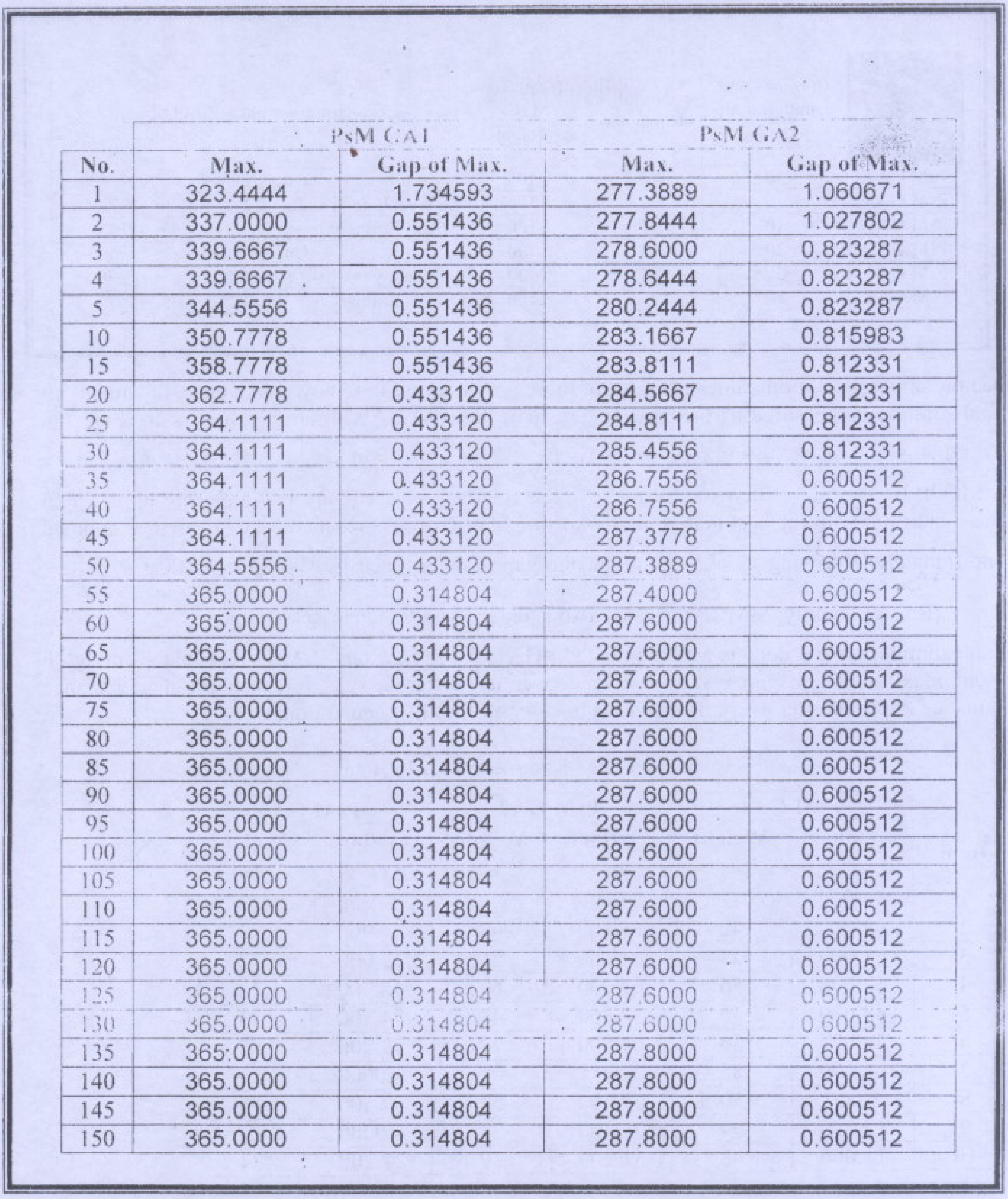
Since the optimal solution values for most of these problems are not know in advance, the quality of a final solution is measured by the percentage gap of the PsM GA's solution value f with respect to the optimal value of the NP-relaxed problem f_{max}^{LP}. This upper bound can easily be determined for any MCPP by sorting all items according to their relative values v_j / w_j and summing up the item value v_j starting, with the best item until a total weight of CW_{max} is reached. The last item is counted proportionately. Knowing the LP optimum, the gap is determined by %-gap = %100(f - f_{max}^{LP} / f_{max}^{LP}). [Rai98]. Finally, MMCP for PsM GA6 is reduced to 0/1 Knapsack Problem.

As an example, table 3 depicts MCPPs for PsM GA1, PsM GA2, and PsM GA6.Table 4 through 6 Present maximum, and gap results averaged over ten different runs for each problem instance (results are depicted after every fifth generation after all first five generations).

Table (3) Tested Knapsack with 14 items

	PsM GA1		PsM GA2		PsM GA6	
Item	Value v_i	Weight w_i	Value v_i	Weight w_i	Value v_i	Weight w_i
1	46.000	14	84.000	70	13.200	11
2	94.000	42	52.800	48	31.900	29
3	61.000	34	61.600	56	12.000	12
4	52.000	56	67.000	67	38.000	38
5	48.000	59	37.800	42	74.700	83
6	44.000	56	64.800	72	79.200	88
7	64.000	85	5.400	6	9.0000	10
8	8.000	50	57.600	64	67.500	75
9	15.000	94	22.400	28	4.0000	5
10	11.000	88	31.200	39	27.200	34

Table (4) Experimental Results of PsM GA1 and PsM GA2



	PsM GA1		PsM GA2	
No.	Max.	Gap of Max.	Max.	Gap of Max.
1	323.4444	1.734593	277.3889	1.060671
2	337.0000	0.551436	277.8444	1.027802
3	339.6667	0.551436	278.6000	0.823287
4	339.6667	0.551436	278.6444	0.823287
5	344.5556	0.551436	280.2444	0.823287
10	350.7778	0.551436	283.1667	0.815983
15	358.7778	0.551436	283.8111	0.812331
20	362.7778	0.433120	284.5667	0.812331
25	364.1111	0.433120	284.8111	0.812331
30	364.1111	0.433120	285.4556	0.812331
35	364.1111	0.433120	286.7556	0.600512
40	364.1111	0.433120	286.7556	0.600512
45	364.1111	0.433120	287.3778	0.600512
50	364.5556	0.433120	287.3889	0.600512
55	365.0000	0.314804	287.4000	0.600512
60	365.0000	0.314804	287.6000	0.600512
65	365.0000	0.314804	287.6000	0.600512
70	365.0000	0.314804	287.6000	0.600512
75	365.0000	0.314804	287.6000	0.600512
80	365.0000	0.314804	287.6000	0.600512
85	365.0000	0.314804	287.6000	0.600512
90	365.0000	0.314804	287.6000	0.600512
95	365.0000	0.314804	287.6000	0.600512
100	365.0000	0.314804	287.6000	0.600512
105	365.0000	0.314804	287.6000	0.600512
110	365.0000	0.314804	287.6000	0.600512
115	365.0000	0.314804	287.6000	0.600512
120	365.0000	0.314804	287.6000	0.600512
125	365.0000	0.314804	287.6000	0.600512
130	365.0000	0.314804	287.6000	0.600512
135	365.0000	0.314804	287.8000	0.600512
140	365.0000	0.314804	287.8000	0.600512
145	365.0000	0.314804	287.8000	0.600512
150	365.0000	0.314804	287.8000	0.600512

Table (5) Experimental Results of PsM GA3 and PsM GA4

	PsM GA3		PsM GA4	
No.	Max.	Gap of Max.	Max.	Gap of Max.
1	271.3778	1.952322	599.4444	2.156788
2	272.2000	1.952322	614.8889	2.125369
3	273.0222	1.952322	624.1111	2.125369
4	275.5111	1.952322	629.4444	1.764054
5	276.5111	1.90437	639.4444	1.041424
10	282.6667	1.767365	642.8889	1.041424
15	284.9111	1.657761	646.8889	1.041424
20	288.0889	1.657761	646.8889	1.041424
25	290.8222	1.000137	646.8889	1.041424
30	290.8222	1.000137	649.8889	0.617272
35	290.8222	1.000137	649.8889	0.617272
40	292.7111	1.000137	649.8889	0.617272
45	294.6000	1.000137	651.0000	0.617272
50	294.6000	1.000137	652.0000	0.617272
55	294.6000	1.000137	653.0000	0.570144
60	294.6000	1.000137	653.0000	0.570144
65	294.6000	1.000137	653.0000	0.570144
70	295.2000	1.000137	653.4444	0.570144
75	295.2000	1.000137	653.4444	0.570144
80	295.2000	1.000137	653.4444	0.570144
85	295.2000	1.000137	653.7778	0.570144
90	295.2000	1.000137	654.2222	0.570144
95	295.2000	1.000137	655.1111	0.570144
100	295.2000	1.000137	655.1111	0.570144
105	295.2000	1.000137	655.3333	0.570144
110	295.2000	1.000137	657.3333	0.570144
115	295.2000	1.000137	657.3333	0.570144
120	295.2000	1.000137	657.5556	0.570144
125	295.2000	1.000137	657.5556	0.570144
130	295.2000	1.000137	657.5556	0.570144
135	295.2000	1.000137	657.5556	0.570144
140	295.2000	1.000137	657.5556	0.570144
145	295.2000	1.000137	657.6667	0.570144
150	295.2000	1.000137	657.6667	0.570144

Table (6) Experimental Results of PsM GA5 and PsM GA6

	PsM GA5		PsM GA6	
No.	Max.	Gap of Max.	Max.	Gap of Max.
1	709.3778	2.529700	97.54445	0.634317
2	716.9000	2.407840	98.32222	0.634317
3	718.8111	2.407840	98.32222	0.634317
4	722.0778	2.407840	98.32222	0.634317
5	728.2222	2.407840	98.76667	0.634317
10	745.0667	1.999608	99.10000	0.634317
15	756.9333	1.999608	99.10000	0.634317
20	763.7222	1.999608	99.10000	0.634317
25	764.7000	1.993515	99.10000	0.634317
30	770.3444	1.810724	99.10000	0.634317
35	772.1667	1.747357	99.10000	0.634317
40	776.0111	1.747357	99.10000	0.634317
45	776.0111	1.747357	99.10000	0.634317
50	776.0222	1.746138	99.10000	0.634317
55	777.2222	1.746138	99.10000	0.634317
60	777.7333	1.746138	99.10000	0.634317
65	777.7333	1.746138	99.10000	0.634317
70	778.4667	1.746138	99.10000	0.634317
75	778.4667	1.746138	99.10000	0.634317
80	778.4667	1.746138	99.10000	0.634317
85	780.1889	1.746138	99.10000	0.634317
90	785.6111	1.746138	99.10000	0.634317
95	786.4778	1.651087	99.10000	0.634317
100	786.4778	1.651087	99.10000	0.634317
105	786.4778	1.651087	99.10000	0.634317
110	786.4778	1.651087	99.10000	0.634317
115	786.4778	1.651087	99.10000	0.634317
120	787.5667	1.531664	99.10000	0.634317
125	787.5667	1.531664	99.10000	0.634317
130	787.5667	1.531664	99.10000	0.634317
135	787.5667	1.531664	99.10000	0.634317
140	788.3667	1.531664	99.10000	0.634317
145	788.3667	1.531664	99.10000	0.634317
150	788.3667	1.531664	99.10000	0.634317

CONCLUSION

As presented in the Previous results, one can see that as correlation between item weight and value is increased, i.e., for example become strongly correlated. The PsM GA evolves slowly toward finding better solutions. It may requires more number of generations to enhance their individuals. Results may also be enhanced using other types of crossover. For example two – point crossover, or even one can hybridize with special local improvement operators to improves results as possible. Also, we can draw the following points:

- Diploidy representation can increase the ability of genetic algorithm toward finding better solutions to the problem in hand. Both diploid / dominance mechanisms and assistant