Clustered Genetic Algorithm to solve Multidimensional Knapsack Problem

Dr. Prabha Shreeraj Nair
Dean Research
Tulsiramji Gayakwade Patil College of Engineering and Technology, Nagpur

ABSTRACT

Genetic Algorithm (GA) has emerged as a powerful tool to discover optimal for multidimensional knapsack problem (MDKP). Multidimensional knapsack problem has recognized as NP-hard problem whose applications in many areas like project selection, capital budgeting, loading problems, cutting stock etc. Attempts has made to develop cluster genetic algorithm (CGA) by mean of modified selection and modified crossover operators of GA. Clustered genetic algorithm consist of (1) fuzzy roulette wheel selection for individual selection to form the mating pool (2) A different kind of crossover operator which employ hierarchical clustering method to form two clusters from individuals of mating pool. CGA performance has examined against GA with respect to 30 benchmark problems for multi-dimensional knapsack. Experimental results show that CGA has significant improvement over GA in relation to discover optimal and CPU running time. The data set for MDKP is available at http://people.brunel.ac.uk/ mastjjb/jeb/orlib/files/mknap2.txt

Keywords: Multidimensional knapsack problem, Genetic algo-rithm, Fitness function, Crossover, Mutation

1 INTRODUCTION

Multidimensional knapsack problem is one of the famous problem in applied mathematics belongs to NP- hard combi-natorial optimization [1]. This problem modeling has applied in several project selection and capital budgeting areas [2-4]. Meirier developed more mechanisms from real word practice to combine models of capital budgeting with novels, and practically applicable technique for evaluation of project [5]. They developed scenario dependent capital budgeting model that have MDKP as a subproblem associated with constraint of generalized upper bound (GUB). Capital budgeting emerged as key challenge for not for profits multihospital healthcare units in United States. D.N. Kleinmuntz and C.E. Kleinmuntz developed a framework for capital budeting which used MDKP formulations [6]. MDKP have been employed to model prob-lems like investment policy in tourism area of developing country [7], database allocation and processor allocation in distributed computing environments [8], groceries delivery in vehicles with multiple compartments [9], cutting stock [10], loading problems [11] and approval voting [12]. Currently, MDKP has been applied in modeling of daily management of remote satellite like SPOT, which consisted in deciding each day what photographs will be attempted the next day [13].

Multidimensional knapsack problem is described by n objects and m knapsacks. Each knapsack has a capacity d_j (j = 1; 2; 3:::m). A number of binary variables x_i (i = 1; 2; 3::n) are used, that is set to 1 if i^{th} object is chosen to inset in knapsacks otherwise it is set to 0. Each object has a profit value p_i (i = 1; 2; 3::n) and a weight w_{ij} corresponding to knapsacks. Thus mathematically this optimization problem can be modeled as :

$$\text{Maximize} \quad \sum_{x_i=1}^{n} p_i x_i \qquad (1)$$

Subject to :

$$\sum_{x_i=1}^{M} w_{ij} x_i \quad d_j \qquad j = 1; 2; :::m \qquad (2)$$

$$x_i \; 2 \; f0; 1g \qquad i = 1; 2; 3::n \qquad (3)$$

The goal of MDKP is to insert a subset of objects into knapsacks that obtain maximum profit without capacity viola-tion of knapsacks. This characteristic reveal this problem has all nonnegative entries. More specifically, we can say $p_i > 0$,

$$d_j > 0,\ 0 \quad w_{ij} \quad d_j \text{ and } \quad w_{ij} \sum_{i=1}^{m} \quad d_j \text{ for all } i\ 2\ n \text{ and } j\ 2\ m$$

Rest of the paper is structured as follows. Genetic algorithm for MDKP is described in Section II. Section III give the overview of related work for MDKP in domain of genetic algorithm. Section IV explain CGA to solve multidimensional knapsack problem. Experimental environments and results are given in Section V. At last conclusion is made in section VI.

II. GENETIC ALGORITHM FOR MULTIDIMENSIONAL KNAPSACK PROBLEM

Genetic algorithm is computation optimization technique which mimic the evolutionary principle i.e. fittest individual will survive in each generation. GA is guided by three main operation: selection, crossover and mutation. Individuals re-production are governed by crossover and mutation operation. Selection operation choose fittest individuals from population to form mating pool. Algorithm run as follow : A set of random individuals are produced which serve as the initial population of GA. The individual can be expressed in real number, binary number, integer number, character based on nature of problem. MDKP uses the binary representation. Individual goodness measurement is estimated through fitness function. MDKP fitness function is maximize the objective function.

Algorithm 1 Genetic Algorithm (GA)

Initialize the control parameters ;

Initialize randomly N individuals as initial population P

(G); while stopping criterion(s) doesn't meet do

i. Choose individuals to create mating pool via selection operator.
ii. Produce new individuals via crossover operator.
iii. Mutate obtained individuals using mutation operator.
iv. Evaluate fitness value of new individuals.
v. Add new individuals with population.
vi. Sort population with respect to fitness value.
vii. Choose top N individuals as population of next generation.

end while

Return the fittest individual as the solution.

Every individual are transformed from one generation to next generation depending on fitness value i.e individual with high fitness value has high probability to participate in upcoming generation. GA employee crossover operator for exploration and mutation operator for exploitation. MDKP uses one point crossover and random mutation. Number of crossover operations are govern through crossover rate while number of mutation operations are govern through mutation rate. A cycle of GA comprise of applying GA operations i.e. evaluation, reproduction, crossover and mutation. Each generation produce a set of individuals. GA termination return the fittest individual as solution. The details of GA operators i.e selection, crossover, and mutation are explained in [14].

III. RELATED WORK

J.P. Martin and C.B Neto and M.K. Crocomo made compar-ison among four linkage learning based genetic algorithm for multidimensional knapsack problem namely, extended com-pact genetic algorithm (eCGA), bayesian optimization algo-rithm (BOA) with detection graphs, BOA with community detection and linkage tree genetic algorithm (LTGA) [15]. eCGA and LTGA have better exploration and exploitation capabilities than BOA versions with small size population (N=100) and produce better quality solutions for MDKPs. All algorithms with large population size produce approxi-mately similar quality solutions for MDKPs but these have different running time and function evaluations. When con-sider solutions quality, functions evaluations and running time of algorithms, BOA versions have better running time than eCGA and LTGA. It is very difficult to make comparison for computational time between eCGA and LTGA because these have different reproduction operators.

Khuri, Back and Heitkotter proposed a GA for 0/1 MDKP [16]. Algorithm allowed breeding and participation of infeasi-ble solutions during search. In this GA, reproduction proceeds by combining the partial information from all element of population. A test suite with few problems were used to test this GA; only moderate quality solutions were produced. Hoff, Lokentangen and Mittet proposed a GA with proper tuned parameters and search mechanisms in which only feasible solutions were allowed during search [17]. This GA discovered optimal for 54 problems out a suite of 55 problems.

Rudolph and Sprave proposed a GA in which selection of parent is not unrestricted as in standard GA but is restricted be-tween neighboring solutions [18]. In their GA every infeasible solutions were penalized as GA of Khursi, Back and Heitkotter [16]. The premier component of this GA be solutions local interaction with in a spatial structured population and self ad-justed controlling mechanism of selection pressure. P. C. Chu and J.E. Beasley proposed heuristic dependent GA for MDKP [19]. In their GA heuristic operator was employed which work with problem specific knowledge. Heuristic operator repaired infeasible solutions and algorithm discover optimal solution among feasible solutions. C. Cota and J.M. Troya proposed another GA which have an improvement mechanism whose objective is to convert infeasible solutions in search space to feasible solutions [20]. Experimental results were better to basic GA, but not as good as Chu and Beasley's GA results.

IV. CLUSTERED GENETIC ALGORITHM FOR MULTIDMENSIONAL KNAPSACK PROBLEM

In this work, clustered genetic algorithm is developed for multidimensional knapsack problem. CGA is explained in algorithm 2. CGA uses an advanced selection method which employee fuzzy and roulette wheel selection called fuzzy roulette wheel selection. Selection mechanism always choose first individual and assigned it to first partition of roulette wheel. Partition size is relative to fitness value. Next similarity value between first individual and remaining individuals are estimated. The individual with highest similarity value occupy the second partition of roulette wheel. This procedure repeat until all individuals are allocated to partition in roulette wheel. Figure 1 show the 4 individuals in binary representation and procedure of assigning individual to the partition of roulette wheel. Then mechanism determined overlapping area between adjacent individuals. Overlapping area size is correlated by similarity value between adjacent individuals. In mating pool formation, a random number is generated. After for each individual membership degree to which generated number belongs is calculated. Individual with highest membership degree is selected. This process repeat until mating pool size is equal to population size. Figure 2 show the fuzzy roulette wheel selection mechanism for 4 individuals in Figure 1.

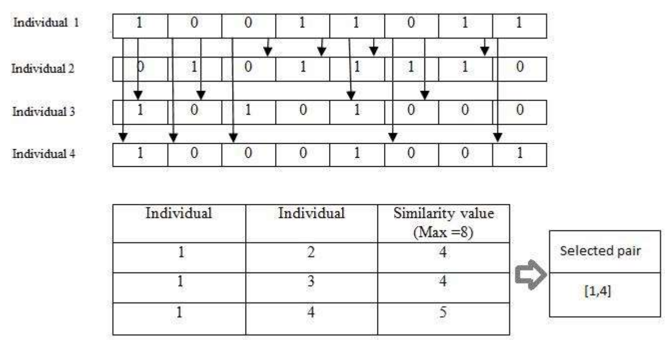

Fig. 1: Procedure of assigning individual to partition of roulette wheel .

CGA adopt a special crossover operator which uses the hierarchical algorithm to form clusters. Individuals in the mating pool are divided into two clusters using hierarchical clustering algorithm. In the hierarchical algorithm, single linkage procedure given in eq (2) is apply to estimate similarity between the two clusters.

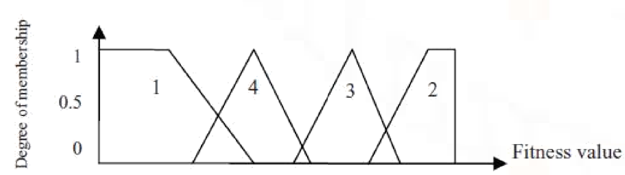

Fig. 2: Fuzzy roulette wheel.

S(X; Y) = M AX fs(x; y)g (4)

x2X;y2Y

Where X and Y are two clusters and s(x,y) stand for similarity between individuals x and y. A pair of offsprings are generated by mating of a random parent from first cluster and a random parent from second cluster. One point crossover operator is employed. This step is continued N/2-1 times. Random mutation is performed for each offspring to keep diversity and to defeat the problem of premature convergence.

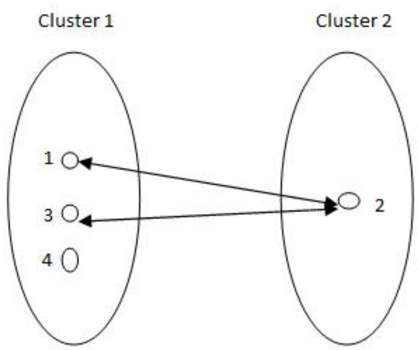

Fig. 3: Selection of parent pair for crossover.

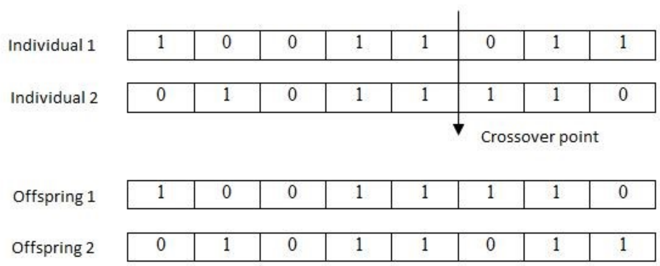

Fig. 4: One point crossover.

V. EXPERIMENTAL RESULTS AND DISCUSSION

To show CGA is better technique to GA, experiments has carried out over 30 standard problems for multidimensional knapsack.

Algorithm 2 Clustered Genetic Algorithm (CGA)

Initialize the control parameters ;

Initialize randomly N individuals as initial population P (G); while stopping criterion(s) doesn't meet do

i. Choose individuals to create mating pool via fuzzy roulette wheel selection operator.
ii. Create two clusters of individuals using hierarchical clustering algorithm .
iii. (iii)Produce new individuals by using crossover operator with a random parent from cluster 1 and a random parent from cluster 2.
iv. Mutate obtained individuals using mutation operator.
v. Evaluate fitness value of new individuals.
vi. Add new individuals with population.
vii. Sort population with respect to fitness value.
viii. Choose top N individuals as population of next generation.

end while

Return the fittest individual as the solution.

A. Experimental Environment

GA and CGA depict in previous section were imple-mentd for multidimensional Knapsack. Experiments were de-signed to estimate average mean error and average mean execution time for GA and CGA. Data set for multidimen-sional knapsack is available at http://people.brunel.ac.uk/ mas-tjjb/jeb/orlib/files/mknap2.txt. Crossover rate CR = 0.7 and mutation rate MR = 0.01. The experiments were conducted on 7th generation intel core i-7-7700k processor (4.2 GHz) and coded in c++ whose compilation is done using Dev c++ compiler.

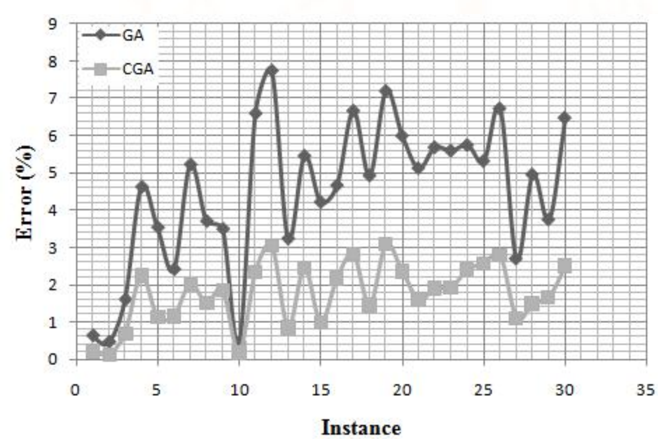

Fig. 5: Error comparision between GA and CGA.

B. Results and Analysis

Experimental results of GA and CGA for MDKPs are given in Table I and Table II respectively. Column 1 denote instance no and column 2 represent MDKP, column 3 and 4 show number of knapsacks and number of objects corresponding to problem, columns 5 denotes the optimal value, columns 6-10 keeps the output value of five run, simulation statical information is given in columns 11-14 and column 15 show mean time taken in execution. GA and CGA has average mean error 4.49% and 1.764% respectively while GA and CGA has average mean execution time 2.34 seconds and 2.34 seconds respectively. CGA has considerable less average mean error in comparison of GA but CGA has slightly higher average mean execution time in comparison of GA. Figure 5 plot the average mean error against the problem instance and Figure 6 plot the average mean execution time against the problem instance.

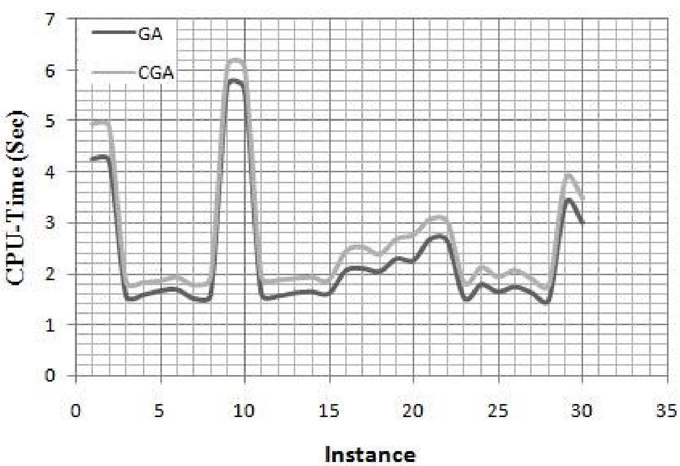

Fig. 6: CPU running time comparison between GA and CGA.

VI. CONCLUSION AND FUTURE WORK

This paper addressed multidimensional knapsack problem via the cluster genetic algorithm to find optimal. CGA employ the fuzzy roulette selection mechanism for individual selection to form mating pool. CGA adopts a special kind of crossover operator in which uses hierarchical clustering algorithm to form two clusters from mating pool. A pair of offspring is generated by mating of a random parent from first cluster and a random parent from second cluster. CGA performance has examined against GA w.r.t 30 standard MDKPs over two criterion i.e average mean error and average mean execution time. As GA and CGA experimental results are shown, CGA has better performance than GA for MDKP. A comparative study of different mutation operators with CGA will be studied in the future work.

REFERENCES

1) Weingartner, H.M. and Ness, D.N., 1967. Methods for the solution of the multidimensional 0/1 knapsack problem. Operations Research, 15(1), pp.83-103.

2) Petersen, C.C., 1967. Computational experience with variants of the Balas algorithm applied to the selection of R&D projects. Management Science, 13(9), pp.736-750.

3) Weingartner, H.M., 1974. Mathematical programming and the analysis of capital budgeting problems. London: Kershaw.

4) McMillan, C. and Plaine, D.R., 1973. Resource allocation via 01 pro-gramming. Decision Sciences, 4, pp.119-132.

5) Meier, H., Christofides, N. and Salkin, G., 2001. Capital budgeting under uncertaintyan integrated approach using contingent claims analysis and integer programming. Operations Research, 49(2), pp.196-206.

6) Kleinmuntz, D.N. and Kleinmuntz, C.E., 2001. Multiobjective capital budgeting in not-for-profit hospitals and healthcare systems. unpublished paper, October.

7) Gearing, C.E., Swart, W.W. and Var, T., 1973. Determining the opti-mal investment policy for the tourism sector of a developing country. Management science, 20(4-part-i), pp.487-497.

8) Gavish, B. and Pirkul, H., 1982. Allocation of databases and processors in a distributed computing system. Management of Distributed Data Processing, 31, pp.215-231.

9) Chajakis, E. and Guignard, M., 1992, October. A model for delivery of groceries in vehicle with multiple compartments and Lagrangean approx-imation schemes. In Proceedings of Congreso Latino Ibero-Americano de Investigacin de Operaciones e Ingeniera de Sistemas.

10) Gilmore, P.C. and Gomory, R.E., 1966. The theory and computation of knapsack functions. Operations Research, 14(6), pp.1045-1074.

11) Bellman, R., 1957. Dynamic Programming Princeton University Press Princeton. New Jersey Google Scholar.

12) Straszak, A., Libura, M., Sikorski, J. and Wagner, D., 1993. Computer-assisted constrained approval voting. Group Decision and Negotiation, 2(4), pp.375-385.

13) Vasquez, M. and Hao, J.K., 2001. A logic-constrained knapsack formu-lation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. Computational Optimization and Applications, 20(2), pp.137-157.

14) Whitley, Darrell. ”An executable model of a simple genetic algorithm.” Foundations of genetic algorithms 2, no. 1519 (2014): 45-62.

15) Martins, J.P., Neto, C.B., Crocomo, M.K., Vittori, K. and Delbem, A.C., 2013, June. A comparison of linkage-learning-based genetic algorithms in multidimensional knapsack problems. In Evolutionary Computation (CEC), 2013 IEEE Congress on (pp. 502-509). IEEE.

16) Khuri, S., Bck, T. and Heitktter, J., 1994, April. The zero/one multiple knapsack problem and genetic algorithms. In Proceedings of the 1994 ACM symposium on Applied computing (pp. 188-193). ACM.

17) Hoff, A., Lkketangen, A. and Mittet, I., 1996, November. Genetic algorithms for 0/1 multidimensional knapsack problems. In Proceedings Norsk Informatikk Konferanse (pp. 291-301).

18) Rudolph, G. and Sprave, J., 1996, September. Significance of locality and selection pressure in the grand deluge evolutionary algorithm. In International Conference on Parallel Problem Solving from Nature (pp. 686-695). Springer Berlin Heidelberg.

19) Chu, P.C. and Beasley, J.E., 1998. A genetic algorithm for the multidi-mensional knapsack problem. Journal of heuristics, 4(1), pp.63-86.

20) Cotta, C. and Troya, J.M., 1998. A hybrid genetic algorithm for the 01 multiple knapsack problem. In Artificial neural nets and genetic algorithms (pp. 250-254). Springer Vienna.

Sr no	**Test Problems**	**m**	**n**	**Optimal**		**Number of Runs**					**Statistical Information**				**Time (s)**
					1	2	3	4	5	Max		Min	Avg	% Err	
1	SENT01	30	60	7772	7712	7736	7725	7719	7723	7736		7712	7723	0.63	4.27
2	SENT02	30	60	8722	8690	8662	8702	8669	8682	8702		8662	8681	0.46	4.23
3	WEING 1	2	28	141278	137902	138804	138602	139903	139904	139904		137902	139023	1.60	1.58
4	WEING 2	2	28	130883	124453	125752	124973	123989	124983	125752		123989	124830	4.62	1.60
5	WEING 3	2	28	95667	92296	91998	92292	92398	92451	92451		91998	92287	3.53	1.68
6	WEING 4	2	28	119337	116625	116441	115998	116459	116792	116792		115998	116463	2.41	1.71
7	WEING 5	2	28	98796	93667	93662	93442	93778	93631	93778		93442	93636	5.22	1.53
8	WEING 6	2	28	130623	124450	126093	126092	126172	126158	126172		124450	125793	3.70	1.59

9	WEING7	2	105	1095445	1054402	1057806	1059803	1057804	1056405	1059803		1054402	1057244	3.49	5.72
10	WEING8	2	105	624319	621422	621798	620642	621882	621761	621882		620642	621501	0.45	5.64
11	WEISH01	5	30	4554	4214	4298	4288	4178	4292	4298		4178	4254	6.59	1.62
12	WEISH02	5	30	4536	4197	4182	4178	4181	4187	4197		4178	4187	7.74	1.58
13	WEISH03	5	30	4115	3990	3987	3976	3973	3984	3990		3973	3982	3.23	1.64
14	WEISH04	5	30	4561	4314	4317	4323	4302	4304	4323		4302	4312	5.45	1.66
15	WEISH05	5	30	4514	4311	4309	4388	4303	4309	4388		4303	4324	4.21	1.63
16	WEISH06	5	40	5557	5331	5333	5307	5228	5291	5333		5228	5298	4.66	2.08
17	WEISH07	5	40	5567	5186	5193	5177	5244	5180	5244		5177	5196	6.66	2.12
18	WEISH08	5	40	5605	5302	5368	5352	5214	5309	5368		5214	5329	4.92	2.06
19	WEISH09	5	40	5246	4901	4848	4898	4886	4812	4901		4812	4869	7.19	2.31
20	WEISH10	5	50	6339	5916	5923	5978	5981	5972	5981		5916	5954	5.99	2.28
21	WEISH11	5	50	5643	5387	5362	5324	5336	5361	5387		5324	5354	5.12	2.70
22	WEISH12	5	50	6339	5984	5972	5988	5977	5974	5988		5972	5979	5.68	2.66
23	HP1	4	28	3418	3220	3198	3224	3278	3215	3278		3198	3227	5.59	1.54
24	HP2	4	35	3186	3016	3010	3078	2999	2912	3078		2912	3003	5.74	1.80
25	PB1	4	27	3090	2916	2924	2942	2918	2930	2942		2916	2926	5.31	1.66
26	PB2	4	34	3186	2978	2986	2962	2971	2963	2986		2962	2972	6.72	1.76
27	PB4	2	29	95168	92372	92708	92504	92666	92780	92780		92372	92606	2.69	1.64
28	PB5	10	20	2139	2037	2032	2043	2012	2041	2043		2012	2033	4.95	1.50
29	PB6	30	40	776	706	740	765	758	766	766		706	747	3.74	3.40
30	PB7	30	37	1035	977	992	966	973	932	992		932	968	6.47	3.02

TABLE I: Result of Genetic Algorithm For Multidimensional Knapsack Problems.

Sr no	Test Problems	m	n	Optimal		Number of Runs					Statistical Information				Time (s)
					1	2	3	4	5	Max		Min	Avg	%Err	
1	SENT01	30	60	7772	7762	7746	7758	7752	7747	7762		7746	7754	0.23	4.95
2	SENT02	30	60	8722	8712	8707	8705	8716	8710	8716		8705	8710	0.14	4.89
3	WEING1	2	28	141278	140177	140183	140196	140178	140771	140771		140177	140301	0.69	1.83
4	WEING2	2	28	130883	127862	127804	128108	127903	127993	127993		127804	1279934	2.25	1.82
5	WEING3	2	28	95667	94570	94582	94448	94684	94556	94582		94448	94568	1.15	1.85
6	WEING4	2	28	119337	117982	117947	117892	117932	117942	117982		117892	117939	1.17	1.92
7	WEING5	2	28	98796	96694	96702	96884	96879	96861	96879		96694	96804	2.02	1.77
8	WEING6	2	28	130623	128503	128664	128704	128630	128639	128704		128503	128628	1.53	1.91
9	WEING7	2	105	1095445	1074998	1076681	1075662	1074552	1073902	1076681		1073902	1075159	1.85	6.10
10	WEING8	2	105	624319	623110	622990	623184	623208	623113	623208		622990	623121	0.19	6.08
11	WEISH01	5	30	4554	4428	4462	4448	4436	4461	4462		4428	4447	2.35	1.93
12	WEISH02	5	30	4536	4402	4398	4372	4410	4403	4403		4372	4397	3.06	1.87
13	WEISH03	5	30	4115	4083	4076	4083	4071	4082	4083		4071	4079	0.87	1.90
14	WEISH04	5	30	4561	4461	4448	4457	4434	4445	4461		4434	4449	2.46	1.92
15	WEISH05	5	30	4514	4476	4487	4462	4462	4468	4487		4462	4452	1.00	1.87
16	WEISH06	5	40	5557	5441	5436	5392	5446	5463	5463		5392	5436	2.18	2.44
17	WEISH07	5	40	5567	5395	5407	5437	5437	5414	5437		5395	5392	2.83	2.52
18	WEISH08	5	40	5605	5518	5507	5532	5522	5536	5536		5507	5523	1.46	2.38
19	WEISH09	5	40	5246	5076	5084	5102	5078	5080	5102		5076	5084	3.09	2.68
20	WEISH10	5	50	6339	6201	6198	6182	6180	6184	6201		6180	6189	2.37	2.76
21	WEISH11	5	50	5643	5558	5537	5529	5568	5568	5568		5529	5552	1.61	3.07
22	WEISH12	5	50	6339	6236	6199	6214	6208	6233	6236		6199	6218	1.91	3.02

23	HP1	4	28	3418	3368	3332	3357	3362	3341	3368		3332	3352	1.93	1.82
24	HP2	4	35	3186	3109	3112	3106	3104	3114	3114		3104	3109	2.42	2.12
25	PB1	4	27	3090	3012	3001	3017	3005	3015	3017		3001	3010	2.58	1.93
26	PB2	4	34	3186	3094	3088	3116	3098	3089	3098		3088	3097	2.79	2.06
27	PB4	2	29	95168	94207	93998	94106	94103	94111	94207		93998	94105	1.12	1.89
28	PB5	10	20	2139	2099	2111	2104	2106	2115	2115		2099	2107	1.50	1.76
29	PB6	30	40	776	763	758	759	766	769	769		758	763	1.68	3.87
30	PB7	30	37	1035	1008	1017	1012	1006	1002	1017		1002	1009	2.51	3.49

TABLE II: Result of Clustered Genetic Algorithm for Multidimensional Knapsack Problems.