Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management
Istanbul, Turkey, July 3 – 6, 2012

Quadratic Multiple Knapsack Problem with Setups and a Solution Approach

Nilay Dogan, Kerem Bilgiçer and Tugba Saraç
Industrial Engineering Department
Eskisehir Osmangazi University
Meselik 26480, Eskisehir Turkey

Abstract

In this study, the quadratic multiple knapsack problem that include setup costs is considered. In this problem, if any item is assigned to a knapsack, a setup cost is incurred for the class which it belongs. The objective is to assign each item to at most one of the knapsacks such that none of the capacity constraints are violated and the total profit is maximized. QMKP with setups is NP-hard. Therefore, a GA based solution algorithm is proposed to solve it. The performance of the developed algorithm compared with the instances taken from the literature.

Keywords
Quadratic Multiple Knapsack Problem with setups, Genetic algorithm, Combinatorial Optimization.

1. Introduction
The knapsack problem (KP) is one of the well-known combinatorial optimization problems. There are different types of knapsack problems in the literature. A comprehensive survey which can be considered as a through introduction to knapsack problems and their variants was published by Kellerer et al. (2004). The classical KP seeks to select, from a finite set of items, the subset, which maximizes a linear function of the items chosen, subject to a single inequality constraint. In many real life applications it is important that the profit of a packing also should reflect how well the given items fit together. One formulation of such interdependence is the quadratic knapsack problem. The Quadratic Knapsack Problems (QKP) asks to maximize a quadratic objective function, subject to a single capacity constraint. QKP has been introduced firstly and the first branch-and-bound algorithm using the bounds based on upper planes has been presented by Gallo et al. (1980). In the proposed branch & bound algorithm for QKP by Chaillou et al. (1986), the computation of an upper bound was based on Lagrangean relaxation. Two upper bounds based on Lagrangean decomposition have been presented by Michelon and Veuilleux (1996) for QKP. In the same year, Billionnet and Calmels (1996) introduced a branch-and-cut approach for QKP. An exact algorithm for QKP was developed by Caprara et al. (1999). In 2000, a number of upper bounds for QKP based on semi definite programming were proposed by Helmberg et al. An exact method based on computation of an upper bound by Lagrangean decomposition has been proposed by Billionnet and Soutif (2003). This method allows finding the optimum solution of instances with up to 150 variables whatever their densities are, and with up to 300 variables for medium and low densities. A greedy Genetic Algorithm (GA), with operators that implement the strategies of the two QKP greedy heuristics proposed by Julstrom (2005). By using the greedy GA, near optimal solutions with very small gap were obtained for the test instances with 100 and 200 variables in reasonable short time. A survey of upper bounds presented in the literature has been given and the relative tightness of several of the bounds has been shown by Pisinger (2007). In the same year, Xie and Liu (2007) presented a mini-swarm approach for QKP. Sipahioglu and Saraç (2009) examined the performance of the Modified Subgradient (MSG) Algorithm to solve the QKP and they showed that the MSG is a successful algorithm on solving QKP. QMKP extends the QKP with k knapsack, each with its own capacity c_k. In QMKP, there are n items to be assigned to k knapsacks of capacities c_k. Each item j has a weight w_j and profit p_j which is the profit achieved if item j is assigned to any of the knapsacks. p_{ij} is the profit achieved if items i and j are both assigned to the same knapsack and the objective is to assign each item to at most one of the knapsacks such that none of the capacity constraints are violated and the total profit of the items put into knapsacks is maximized. QMKP is NP-Hard as it reduces to QKP when the knapsack number is one. QMKP has been introduced and the first solution approaches have been produced by Hiley and Julstrom (2006). They developed three heuristic approaches; a greedy heuristic, a stochastic hill-climber and a Genetic Algorithm for QMKP. Saraç and Sipahioglu (2007) proposed a genetic algorithm for solving QMKP. Unlike Hiley and Julstrom (2006), there is no assumption that all the knapsack capacities are the same. And it is demonstrated that the GA_{SS} is more successful than the GA_{HJ} on the instances with large number of knapsacks. In the same year, Singh and Baghel (2007) proposed a new steady-state grouping genetic algorithm for the QMKP. Like Hiley and Julstrom (2006), they also assume that all knapsack capacities c_k are

same. They compared their results to two previously proposed methods – the genetic algorithm (Hiley and Julstrom, 2006) and the stochastic hill climber (Hiley and Julstrom, 2006). The results showed the effectiveness of their approach. Average values of results obtained by the proposed algorithm are always better than those obtained with two approaches published by Hiley and Julstrom (2006), the genetic algorithm and the stochastic hill climber. Sundar and Singh (2010) proposed a new variant of the artificial bee colony algorithm for the QMKP. Their proposed algorithm has been compared to four best previous approaches stochastic hill climber (Hiley and Julstrom, 2006), genetic algorithm (Hiley and Julstrom, 2006), genetic algorithm (Saraç and Sipahioglu, 2007) and genetic algorithm (Singh and Baghel, 2007). Computational results show the superiority of their approach over these approaches in terms of solution quality.

In recent years, studies on knapsack problem that takes the setup constraints into consideration have started to be included in the literature (McLay and Jacobson, 2007; Caserta et al., 2008; Altay et al., 2008; Michel et al., 2009). In these problems, when an item is assigned to a knapsack, also the setup cost for the class of it has to be attained to the knapsack. Further, not only the weight of the item, but also the cost for setup has to be taken into consideration in terms of capacity utilization. In the literature, items that require a common setup, rather than separate setups, when being assigned to the same knapsack, are considered as items of the same class or family.
McLay and Jacobson (2007), provides three dynamic programming algorithms that solve The Bounded Setup Knapsack Problem (BSKP) in pseudo-polynomial time and a fully polynomial-time approximation scheme (FPTAS). One of the dynamic programming algorithms presented solves the Bounded Knapsack Problem (BKP) with the same time and space bounds of the best known dynamic programming algorithm for BKP. The FPTAS improves the worst-case time bound for obtaining approximate solutions to BKP as compared to using FPTASs designed for BKP or the 0-1 Knapsack Problem. Caserta et al. (2008), propose a new metaheuristic-based algorithm for the Integer Knapsack Problem with Setups. Proposed algorithm is a cross entropy based algorithm, where the metaheuristic scheme allows to relax the original problem to a series of well chosen standard Knapsack problems, solved through a dynamic programming algorithm. Altay et al. (2008), consider a class of knapsack problems that include setup costs for families of items. A mixed integer programming formulation for the problem is provided along with exact and heuristic solution methods. Computational performance of the algorithms are reported and compared to CPLEX. Michel et al. (2009), consider the multiple-class integer knapsack problem with setups. Their paper provides a review of the literature on knapsack problems with setups, discusses various reformulations, and presents specialized branch-and-bound procedures extending the standard algorithm for the knapsack problem. Sang and Sang (2012), propose a new memetic algorithm for the quadratic multiple container packing problem. The proposed memetic algorithm is based on the adaptive link adjustment evolutionary algorithm (ALA-EA) and it incorporates heuristic fitness improvement schemes into the ALA-EA. Wang et al. (2012), provide a comparison of quadratic and linear representations of QKP based on test problems with multiple knapsack constraints and up to eight hundred variables. Best of our knowledge, there isn't any study on quadratic multiple knapsack problem with setups in the literature. In this study, firstly we presented a mathematical model for the QMKP with setups and then because of the HP-hard nature of the problem, a genetic algorithm based solution approach has been proposed to solve it.

In the next section, a mathematical model for QMKP with setup is presented. Section 3 provides an introduction to the solution algorithm for QMKP with setups in detail. Computational results are presented in Section 4 and conclusions are outlined in Section 5.

2. Mathematical Model of the Quadratic Multiple Knapsack Problem with setups

Notations and the mathematical model of the Quadratic Multiple Knapsack Problem with setups is as follows:

Index sets:

$J = \{j|\ j = 1,..,n\}$ index set of items
$K = \{k|\ k = 1,..,m\}$ index set of knapsacks
$R = \{r|\ r = 1,..,l\}$ index set of class

Parameters:

n : number of items
m : number of knapsacks
l : number of class
q_{ij} : profit achieved if item i and j are assigned to same knapsack
p_j : profit achieved if item j is assigned to any knapsack
c_k : capacity of knapsack k

w_j : weight of item *j*
s_r : setup cost of a item belong to class *r*
t_{rj} : 1, if item *j* is belong to class *r*, 0, otherwise
U : a big positive number

Decision variable:

x_{jk} : 1, if item *j* is selected to knapsack *k*, 0, otherwise

y_{rk} : 1, if any item belong to class *r* is selected to knapsack *k*, 0, otherwise

(QMKP with setups):

$$\max \quad z = \sum_{j=1}^{n} p_j x_{jk} + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{m} q_{ij} x_{ik} x_{jk} \tag{1}$$

subject to

$$\sum_{j=1}^{n} w_j x_{jk} + \sum_{r=1}^{l} s_r y_{rk} \le c_k, \quad k = 1,......,m \tag{2}$$

$$\sum_{j=1}^{n} t_{rj} x_{jk} \le U y_{rk}, \quad k = 1,.......,m, \; r = 1,......,l \tag{3}$$

$$\sum_{k=1}^{m} x_{jk} \le 1, \quad j = 1,.......,n \tag{4}$$

$$x_{jk} \in \{0,1\}, \quad j = 1,.......,n, \; k = 1,......,m \tag{5}$$

The objective (1) is maximizing the total profit of the items assigned to knapsacks. Constraints (2) ensure that the capacities of knapsacks are not exceeded. Constraints (3) provide that if any item belongs to class *r* is selected to knapsack *k*, the value of y_{rk} is one. Constraints (4) provide that item *j* can be assigned to only one of the knapsacks. Binary restrictions on decision variables are imposed by constraints (5).

3. Solution Approach

Most of the combinatorial optimization problems are known to be NP-hard and they are quite hard to be solved by conventional optimization techniques. QMKP is one of them. QMKP with setups is also NP-hard by restriction to QMKP; set all the setup values s_r to zero. For that reason, in this study a genetic algorithm based solution approach is proposed for QMKP with setups. Our solution approach has two steps. In the first step, a GA find a solution for the problem and the second step, an algorithm improves the solution obtained the first step. Proposed method is explained below.

In this study, *n* bit string is used, where *n* is the number of the items to be assigned to *k* knapsacks. For instance, a solution to a problem with 10 items and 3 knapsacks can be encoded to a chromosome as [0130201302]. This chromosome gives the information that items 2 and 7 are assigned to the 1st knapsack, items 5 and 10 are assigned to the 2nd knapsack and items 3 and 8 are assigned to the 3rd knapsack. The feasibility of the chromosomes that are generated by the initialization procedure must be satisfied while they are being processed by the genetic operators. Genetic operators are introduced to prevent us from having this infeasibility as explained below.

Reproduction. The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. The chance that a string will be copied is based on the string's fitness value, calculated from a fitness function. In this study, 2-tournament selection method is used as reproduction operators.

Crossover. Crossover enables the algorithm to extract the best genes from different individuals and recombine them into potentially superior children. In this study, to maintain the feasibility of chromosomes, a specialized uniform based crossover operator developed by Saraç and Sipahioglu (2007) is used. In this operator, randomly selected two parents chromosomes are used to generate two offspring. Randomly selected gene of the first chromosome is interchanged to the corresponding gene that is placed in the same order of the second chromosome, if the capacities of the knapsacks are available for both of the chromosomes. If any capacity of knapsack exceed by adding a new gene, came from the other parent chromosome, the value of this gene change as zero.

Mutation. Reproduction and crossover alone can obviously generate a staggering amount of differing strings. However, depending on the initial population chosen, there may not be enough variety of strings to ensure the GA searches the entire problem space, or the GA may find itself converging on strings that are not quite close to the optimum it seeks due to a bad initial population. Some of these problems may be prevented by introducing a mutation operator into the GA. Our mutation operator, firstly, generates a random number and determines the gene. Provided that generated number's value is different from the gene's value, a new random number for the gene which will be made mutation be generated as knapsack numbers. The capacity is controlled. If the capacity is exceeded, the above mutation steps are repeated.

Elitism. When creating a new generation, there is always a risk of losing the most fit individuals. Using elitism, the most fit individuals are copied to the next generation. The other ones undergo the crossover and mutation. Since the elitism selection improves the efficiency of a GA considerably, as it prevents losing the best results, it is used in developed GA.

Improvement. Martello and Toth (1981) proposed a polynomial time approximate algorithm for multiple knapsack problems with linear objective function. We modified the improvement techniques of this approximate algorithm for QMKP with setups. Improvement algorithm used these techniques explained below as pseudo code.

```
    for j := n to 1 step-1 do if y_j>0 then
        begin
          ć := ć_yj + w_j;
          Y:= Ø;
        for k := 1 to n do
            if y_k=0 and w_k ≤ ć then
                begin
                    Y:= Y ∪ {k};
                    ć := ć - w_k
                end

        if ∑_{k∈y} p_k > p_j then
                    begin
                    for  each k ∈ Y do y_k := y_j;
                    ć_yj := ć;
                    y_j =0;
                    z=z+ ∑_{k∈Y} p_k- p_j
                    end
        end
end
```

ć : the rest capacity of the knapsack
y_j: j^{th} gene's value on chromosome

If there is a gene not assigned to any knapsack on chromosome, it is exchanged with the assigned genes. Then the capacity is controlled. If it is not exceeded, fitness value is calculated and compared the before value. In case the new fitness value is better than the old one, new obtained chromosome is hold.

4. Computational Results

In this section, firstly, the results obtained with developed algorithm for forty two QMKP instances given in the literature are reported and the performance of the developed algorithm is compared to performances of the previous algorithms (Julstrom (2005), McLay (2007), Caserta, (2008)) in terms of solution quality. And then, the setup values are randomly generated for the instances, and QMKP with setups instances are occurred. Also the obtained results with developed GA for forty two QMKP with setups instances are reported.

The algorithm was coded with C# and all computational experiments were carried out on PCs with hardware of 2.67GHZ CPU and 4 GB RAM. The algorithm has some parameters; population size, crossover rate and mutation rate. The population size was chosen as 20. The probability that crossover generated offspring chromosomes was determined as 0.10, and mutation rate were chosen as 0.015 for all tests.

A significant feature of QMKP instance is the density of its linear values p_i and quadratic values p_{ij} which are non-zero. Instances with two different densities (0.25, 0.75), two different numbers of items (100, 200) and three different numbers of knapsacks (3, 5, 10) were solved. All QKP instances are available on the web site (http://cedric.cnam.fr/~soutif/QKP/) and the number of knapsacks and the capacities of the knapsacks, which are exactly the same values with the values used by Hiley and Julstrom (2006), adopted to convert these QKP instances to QMKP instances, are given in Table 1 and Table 2.

Computational results for forty two QMKP instances are illustrated in Table 1 and Table 2. The left part of the tables summarize the features of the instances. For each instance, the table lists its number of knapsacks (*k*), density (*d*), number of items (*n*), instance number given on the web site (*no*) and capacities of the knapsacks (*cap.*). The following parts of Table 1 and Table 2 present the best and mean values for each study ((Julstrom (2005), McLay (2007), Caserta, (2008), Michel (2009)) published in the literature, respectively. Best of *best* results and best *values mean* of each algorithm for each test problem were marked as bold in Table 1 and Table 2. As clearly can be seen from Table 1, generally close results to the previous study are obtained by the proposed algorithm and superior best results in 4 of 30 instances with 0,25 density are also achieved.

Table 1. Performance of the proposed algorithm on the 27 QMKP instance with 0.25 density

features of the instances					GA_{HJ}		GA_{SS}		SHC_{HJ}		GA_{SSG}		*ALA-EA*		*proposed alg.*	
K	*d*	*n*	*no*	*cap.*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*
3	0,25	100	1	688	28665	27904	28807	28514	28144	27635	28798	28485	**30604**	28072.1	28050	27861
3	0,25	100	2	788	28059	27044	28456	28225	**29915**	26222	28036	27507	29587	26934.7	27209	27091
3	0,25	100	3	663	26780	25991	26754	26574	25945	25193	26936	26411	**28304**	26746.9	26387	25810
3	0,25	100	4	804	28199	27265	28383	28035	27109	26127	28418	27473	**29083**	26821	27531	27082
3	0,25	100	5	845	27550	26683	27582	22043	26288	25617	27617	26971	29194	26984.5	**29682**	29650
5	0,25	100	1	413	21914	21315	22039	21735	21584	20911	22038	21662	**23156**	21273.9	20094	19788
5	0,25	100	2	473	21216	20472	21249	20724	20934	19768	21459	21046	**21936**	19466.9	20264	20141
5	0,25	100	3	398	20243	19763	20862	20444	19454	18765	21012	20279	**21512**	19058.3	18837	18591
5	0,25	100	4	482	21698	20923	21601	21417	20173	19730	21987	21344	**22030**	20181.9	20643	20510
5	0,25	100	5	507	20808	20248	20928	16779	19932	18843	21057	20304	21700	19504.7	**21947**	21834
10	0,25	100	1	206	13521	12499	15778	15505	15235	14737	15663	15201	**16277**	14457.1	14082	13701
10	0,25	100	2	237	12859	12019	14835	14601	14210	13684	**15002**	14654	14590	13259.1	13890	13695
10	0,25	100	3	199	11790	11245	14348	14136	13334	12918	14231	13716	**14455**	12579.4	13231	12573
10	0,25	100	4	241	13316	12593	15495	15179	14321	13867	**15979**	15310	15100	13505.5	13980	13685
10	0,25	100	5	254	11909	11389	14770	11666	13405	12929	14510	14018	14821	13126.9	**14922**	14610
3	0,25	200	1	1343	97469	95497	99853	99216	99232	98169	99753	99286	**105550**	99337.7	93082	91131
3	0,25	200	2	1424	106162	100521	104277	101179	106730	105857	107475	107036	**114462**	108265.6	108943	106387
3	0,25	200	4	1506	95649	93968	97700	97525	97407	97067	98276	97092	**102573**	96246.9	97937	96318
3	0,25	200	5	1230	99458	96077	98326	97980	100827	99762	101463	100612	**106669**	99361.7	85945	84570
5	0,25	200	1	805	70731	68705	73619	72600	72277	70776	73040	72216	**75148**	70582.9	63534	61639
5	0,25	200	2	855	76297	72924	74883	74403	77551	76643	78428	77236	**83874**	78860.7	74679	73403
5	0,25	200	4	904	70264	67416	71936	71339	71307	69417	71964	70892	**73512**	68873.1	68838	67358
5	0,25	200	5	738	72745	69978	73825	72006	74287	73229	74936	74538	**76685**	71650.4	62641	60213
10	0,25	200	1	403	42016	39791	48119	47653	48006	46960	**49212**	48065	47415	43148.6	39012	37650
10	0,25	200	2	427	45483	42739	51666	50410	51438	50622	52153	51568	**53002**	49144.2	45114	43970
10	0,25	200	4	452	41623	39446	**48792**	47907	47296	45751	47853	47001	47197	42712.5	44269	42252
10	0,25	200	5	369	46811	42399	49504	48698	50402	49431	**51000**	50267	50838	46381.5	39287	37130

Table 2. Performance of the proposed algorithm on the 15 QMKP instance with 0.75 density

features of the instances					GA_{HJ}		GA_{SS}		SHC_{HJ}		GA_{SSG}		*ALA-EA*		*proposed alg.*	
K	*d*	*n*	*no*	*cap.*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*	*best*	*values mean*
3	0,75	100	1	822	69769	68941	64335	63757	69786	69172	69935	69694	75706	72275.4	**83042**	82395
3	0,75	100	2	714	69146	68639	68164	66585	69056	68508	69344	69203	**73440**	70686.6	68226	67828
3	0,75	100	3	686	68763	67557	67643	66257	68547	67939	68776	68518	**73318**	70675.7	68058	66763
3	0,75	100	4	666	69907	69101	68626	65018	69646	69003	69696	69677	**73787**	71133.8	68958	68503
3	0,75	200	1	1311	268919	265523	261106	254301	269447	267765	269351	268506	**282727**	273885.4	265771	261311
5	0,75	100	1	493	48663	47678	47449	45902	48888	48138	48675	48414	53142	50157.7	**59030**	57915
5	0,75	100	2	428	48990	48175	47766	47032	48686	48028	48916	48376	**51414**	49061	47618	47349
5	0,75	100	3	411	47512	46623	48008	46587	47396	46970	48126	47815	**50524**	47563.4	46447	45245
5	0,75	100	4	400	49845	49194	46921	46063	49468	48864	49724	49297	**52940**	50964.5	48936	47076
5	0,75	200	1	786	179525	177438	173905	170447	182374	181203	183318	182197	**191865**	185667	176072	167250
10	0,75	100	1	247	26603	25681	28767	27723	29136	28665	29179	28762	**31687**	29950.9	30353	29216
10	0,75	100	2	214	28663	27815	29824	29344	30367	30031	30640	30357	**31629**	30016.2	28985	27505
10	0,75	100	3	205	26176	25038	27960	27282	28838	28297	28857	28561	**30584**	28830.3	26677	26159
10	0,75	100	4	200	29701	28592	30712	29135	30624	30346	31039	30581	**33031**	31385.6	28037	27591
10	0,75	200	1	393	102002	98962	106008	100940	110238	109028	110528	109755	**114016**	108874.6	94196	89866

As shown Table 2, like Table 1, close results to the published results in the literature are obtained by using the algorithm presented in this study. The proposed algorithm can also be obtained superior results in 3 of 15 instances with 0,75 density. In addition, we can show that the density of matrix affects the success of the developed algorithm.

Computational results for forty two QMKP with setups instances are illustrated in Table 3. The left part of the table summarizes the features of the instances. The following parts of it present the best and mean values of our algorithm, respectively. As clearly can be seen from Table 3, proposed algorithm is capable to solve the QMKP with setups.

5. Conclusions

In this study, QMKP with setups has been introduced first and a new GA based solution method has also been proposed to solve it. Computational test results for the QMKP instances demonstrated that for 7 of the test problems, the algorithm has achieved the best results reported in the literature. Near results to the published before in the literature are also obtained for the remaining instances and the algorithm is very successful in some instance. It is also remarkable that the proposed algorithm’s result with 0,75 density is better than the result with 0,25 density. Since the effectiveness of the proposed algorithm depends on user-supplied parameters, it is recommended that an analysis to decide on the appropriate values of the parameters of it may be carried out, which is considered as a further research on this subject.

Table 3. Performance of the proposed algorithm on the 42 QMKP with set up instance with 0.25 density

features of the instances					*proposed alg.*	
K	*d*	*n*	*no*	*cap.*	*best*	*values mean*
3	0.25	100	1	688	26888	26782
3	0.25	100	2	788	28008	27332
3	0.25	100	3	663	24967	24967
3	0.25	100	4	804	26726	26373
3	0.25	100	5	845	29552	28798
5	0.25	100	1	413	19938	19719
5	0.25	100	2	473	20436	19939
5	0.25	100	3	398	18225	17730
5	0.25	100	4	482	19897	19626
5	0.25	100	5	507	21286	20964
10	0.25	100	1	206	13734	13349
10	0.25	100	2	237	14062	13711
10	0.25	100	3	199	12362	12036
10	0.25	100	4	241	13493	13174
10	0.25	100	5	254	14033	13795
3	0.25	200	1	1343	92212	90839
3	0.25	200	2	1424	105516	104552
3	0.25	200	4	1506	96892	95903
3	0.25	200	5	1230	84548	82998
5	0.25	200	1	805	63023	61341
5	0.25	200	2	855	73358	72045
5	0.25	200	4	904	67249	66627
5	0.25	200	5	738	60206	58561
10	0.25	200	1	403	39613	38634
10	0.25	200	2	427	45972	44013
10	0.25	200	4	452	41048	28653
10	0.25	200	5	369	39366	36938
3	0.75	100	1	822	77730	77634
3	0.75	100	2	714	67297	66230
3	0.75	100	3	686	66848	66537
3	0.75	100	4	666	68390	67617
3	0.75	200	1	1311	263389	258749
5	0.75	100	1	493	55776	54687
5	0.75	100	2	428	47346	47131
5	0.75	100	3	411	44884	43792
5	0.75	100	4	400	47676	47023
5	0.75	200	1	786	173912	166557
10	0.75	100	1	247	29193	26872
10	0.75	100	2	214	27875	25449
10	0.75	100	3	205	26340	25314
10	0.75	100	4	200	26310	26250
10	0.75	200	1	393	93803	89593

References

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). *Knapsack Problems*. New York: Springer-Verlag.

Gallo, G., Hammer, P. L., & Simeone, B. (1980). Quadratic Knapsack Problems. *Mathematical Programming Study*, 12, 132-149.

Chaillou, P., Hansen, P., & Mahieu, Y. (1986). Best network flow bound for the quadratic knapsack problem. *Combinatorial Optimization of Lecture Notes in Mathematics*, 1403, 225-235.

Michelon, P., & Veuilleux, L. (1996). Lagrangean methods for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, 92, 326-341.

Billionnet, A., & Calmels, F. (1996). Linear programming for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, 92, 310-325.

Caprara, A., Pisinger, D., & Toth, P. (1999). Exact solution of the quadratic knapsack problem. *INFORMS Journal on Computing*, 11, 125-137.

Helmberg, C., Rendl, F., & Weismantel, R. (2000). A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4, 197-215.

Billionnet, A., & Soutif, E. (2003). An exact method based on Lagrangean decomposition for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, 157, 565-575.

Julstrom, B. A. (2005). Greedy, genetic, and greedy genetic algorithms for the quadratic knapsack problem. *Proceedings of the Genetic and Evolutionary Computation Conference*, 1, (pp. 607-614).

Pisinger, D. (2007). The quadratic knapsack problem-a survey. *Discrete Applied Mathematics*, 155, 623-648.

Xie, X. F., & Liu, J. (2007). A Mini-Swarm for the Quadratic Knapsack Problem. *IEEE Swarm Intelligence Symposium (SIS)*, (pp. 190-197), Honolulu, HI, USA.

Sipahioglu, A., & Saraç, T. (2009**).** The Performance of the Modified Subgradient Algorithm on Solving the 0-1 quadratic knapsack problem. *INFORMATICA*, 20(2), 1-12.

Hiley, A., & Julstrom, B. A. (2006). The Quadratic multiple knapsack problem and three heuristic approaches to it. *Proceedings of the Genetic and Evolutionary Computation Conference*, (pp. 547-552).

Saraç, T., & Sipahioglu, A. (2007). A genetic algorithm for the quadratic multiple knapsack problem. *Lecture Notes in Computer Science*, 4729, 490-498.

Singh A., & Baghel A. S. (2007). A New Grouping Genetic Algorithm for the Quadratic Multiple Knapsack Problem. *EvoCOP*, *LNCS 4446*, (pp. 210 – 218).

Sundar, S., & Singh, A. (2010). A Swarm Intelligence Approach to the Quadratic Multiple Knapsack Problem. *Neural Informatıon Processıng: Theory And Algorıthms*, 6443 (pp.626-633).

McLay L.A., Jacobson S.H. 2007. Algorithms for the bounded set-up knapsack problem *Discrete Optimization* 4, 206–212.

Caserta M., Quiñonez Rico E., Márquez Uribe A., "A cross entropy algorithm for the Knapsack problem with setups", *Computers & Operations Research*, 35 (2008) 241 – 252.

Altay N., Robinson Jr P.E.. Bretthauer K.M. 2008. Exact and heuristic solution approaches for the mixed integer setup knapsack problem, *European Journal of Operational Research*, 190 598–609.

Michel S., Perrot N., Vanderbeck F. 2009. Knapsack problems with setups. *European Journal of Operational Research*, Vol.196, pp. 909–918.

Sang M.S. and Sang W.L., "A memetic algorithm for the quadratic multiple container packing problem", Appl Intell (2012) 36:119–135.

Wang H., Kochenberger, G. and Glover, F., "A computational study on the quadratic knapsack problem with multiple constraints", Computers & Operations Research 39 (2012) 3–11.

Martello, S., Toth, P.: Heuristic algorithms for the multiple knapsack problem. Computing 27 (1981) 93-112.

Web Site References: http://cedric.cnam.fr/~soutif/QKP/ (2011).