2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Lagrangian and Surrogate information enhanced tabu search for the MMKP

Skander HTIOUECH
Sys-com
National Engineering School of Tunis
Skander.Htiouech@enib.rnu.tn

Sadok BOUAMAMA
HANA Lab.
National School of Computer Science of Tunisia
Sadok.Bouamama@ensi.rnu.tn

***Abstract*—The multidimensional multi-choice knapsack problem (MMKP) is NP-hard. Within the framework of solving this problem, we suggest newer approaches. We not only propose a multi-starts version of our previous works aproach using surrogate constraint informations based choices [31][32], but also we introduce another newer heuristic. The latter uses Lagrangian relaxation informations in place of surrogate informations. Compared with other literature known methods described so far, our approaches experimentations results are competitive.**

I. INTRODUCTION

The MMKP is a combinatorial optimization problem, one of the most complex members of the knapsack problem family [7]. Let a set N=$\{N_1, \ldots, N_i, \ldots, N_n\}$ of n disjoint item groups, where each group i, i=$1, \ldots, n$ has n_i items. Each item j, j=$1, \ldots n_i$, of the i^{th} group has a non-negative profit value c_{ij}, and requires an amount of resources represented by the weight vector a_{ij}=$(a^1_{ij}, a^2_{ij}, \ldots, a^k_{ij})$. Note that weight terms a^k_{ij} (with $1 \le k \le m, 1 \le i \le n, 1 \le j \le n_i$) are nonnegative. b=$\left[b^1, b^2, \ldots, b^m\right]$ is the capacity vector of the multi-constrained knapsack resources. It is worthy to note that x_{ij} takes either 1 or 0, which means that item j of the i^{th} group is picked or not, respectively.

The goal of the MMKP is to pick exactly one item from each group such that the resource constraints are not violated and the selected items profit is maximized, as well. Formally the problem can be stated as follows :

$$\mathit{Max} \quad \sum_{i=1}^{n} \sum_{j=1}^{n_i} c_{ij} x_{ij} \tag{1}$$

$$\mathit{Subject\ to} \quad \sum_{i=1}^{n} \sum_{j=1}^{n_i} a^k_{ij} x_{ij} \le b^k \ , k = 1, \ldots, m \tag{2}$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \ , i = 1, \ldots, n \tag{3}$$

$$x_{ij} \in \{0, 1\}, i = 1, \ldots, n, j = 1, \ldots, n_i \tag{4}$$

To exclude trivial solutions, we assume that for all $1 \le j \le n_i$, we have :

$$\sum_{i=1}^{n} min\{a^k_{ij}\} \le b^k \le \sum_{i=1}^{n} min\{a^k_{ij}\} \quad k = 1, \ldots, m \tag{5}$$

Many real life problems, such as the telecommunications [2][3], logistics [5], financial sectors [1], military strategy [6][4], could be easily fomalized as an MMKP .

The first approaches described in the literature are derived from multiple choice knapsack problem (MCKP) [15], a special case of MMKP ($m = 1$) [44]. In this problem, a single capacity constraint must be satisfied. The MMKP is also closely related to another non-standard variant, namely the multidimensional knapsack problem (MDKP) [16][17][18][19]. The latter is obtained from MMKP removing choice constraints and considering the number of the $n = 1$ (all objects are in the same group). In fact, the MMKP is considered like a combination of MDKP and MCKP [47][44]. Counter to MCKP and MDKP, little works are devoted to the MMKP [46][48].

Different exact methods are described in the literature for MCKP. Sinha and Zoltners [20] proposed a branch and bound algorithm guided by the solutions of the linear programming relaxation (LP) at each node. Numerical experiments conducted by the authors on random instances show that the computation time increases significantly with the number of groups (or classes) then the number of objects. Armstrong *et al.* [21] have improved this algorithm to reduce the required storage space and computation time for larger instances.

Dyer *et al.* [45] proposed a hybrid algorithm combining dynamic programming and branch and bound to solve the MCKP. The authors use the Lagrangian duality [50] to compute bounds at the branching nodes, and apply a reduction procedure. The results of presented in [45] show the potential of hybridization compared to conventional branch and bound algorithms.

Pisinger [22] describes a partitioning algorithm with polynomial complexity to find an optimal solution of the (LP) linear relaxation. He also discussed the integration of this method in a dynamic programming algorithm based on the enumeration of a minimum number of groups. He proposes a dynamic programming based on that adds (if necessary) groups approach the core problem. With this mechanism, the author improved the results obtained with other algorithms for large instances with more than 100,000 variables.

The first results of the resolution of the MMKP are due to Moser *et al.* [43]. The authors developed a heuristic based on Lagrangian relaxation that starts from a feasible solution, and switches repeatedly objects to reduce its ”non releasability”. Their algorithm was later improved by Akbar *et al.* [23].

Khan *et al.* [44] proposed a heuristic based on aggregation constraints as by suggested Toyoda in [24] to solve the MDKP. They have improved their approach using an object exchange procedure. The heuristic has been compared to a branch and bound algorithm and the approach of Moser *et al.* [43]. The results of the calculations provided in [44] show that the heuristic is considerably better than the other algorithms both in terms of computing time and in terms of quality of solutions.

Parra-Hernandez and Dimopoulos [13] adapted algorithm of Pirkul [25] for MKP to solve the MMKP. Initially, they relax the constraints of choice and thus transform the MMKP MKP with a generalized upper bounds (the choice constraints $\sum_{j=1}^{n_i} x_{ij} = 1$ is transformed into $\sum_{j=1}^{n_i} x_{ij} \leqslant 1$). In other words, more than one object can be selected in each group. One solution (not necessarily feasible) is then obtained and improved by using a local search. Their algorithm produces better solutions than those obtained by Akbar *et al.* [23] penalized with a significant increase in computation time.

Hifi *et al.* [48] developed two constructive heuristics and a method of guided local search to solve the MMKP. Their approaches have led to better results than those obtained by Moser *et al.* [43] and Khan *et al.* [44]. These results were improved by the same authors in [52] using a reactive local search algorithm .

Akbar *et al.* [26] described a heuristic for MMKP based on the production of convex hulls and aggregation constraints multidimensional capabilities in a single penalty using a vector. The authors obtained encouraging results especially for uncorrelated with those appearing in the Moser *et al.* [43] and Akbar *et al.* [23] instances. In [27], Hiremath and Hill describe two greedy heuristics for the MMKP and provide an empirical study of instances of specific tests to show the performance of their algorithms.

More recently, Cherfi and Hifi [51] described a branch and bound algorithm for the MMKP using a variant of the column generation algorithm and a heuristic rounding to assess the connections of nodes. Different branching patterns were explored. Their approaches were compared with CPLEX [28] and heuristics described by Hifi *et al.* [52] on a set of test instances from the literature. For 21 of the 33 instances , the best known lower bounds have been improved.

In [29], Hanafi *et al.* applied three iterative heuristics-based relaxation to solve the MMKP inspired by their previously proposed approaches for Hanafi and Wilbaut [30]. The authors considered two strategies for selecting subsets of variables whose integrity constraints should be released. The lower bounds are computed by solving a reduced problem obtained by fixing some of the variables to their value in the linear optimal solution. The authors compared their approach with algorithms described by Cherfi and Hifi in [51].

By reference to Hanafi *et al.*[29], Crevits *et al.* [33] developed a heuristic based on a new relaxation that removing integrity constraints and forcing variables close to 0 or 1. This relaxation is more general than the PL and mixed integers relaxation [29]. The authors improved their algorithm by integrating a simple descent procedure as a local search that attempts to preserve each iteration of the realizability of solutions. His local search procedure uses the special structure of MMKP based on the exchange of two objects in the same group.

Other metaheuristics have also been recently proposed for the MMKP [34][35]. In [35], Ren and Feng use an ant colony optimization algorithm for the MMKP. Their algorithm combines a constructive process with an efficient operator used to repair infeasible solutions. Iqbal *et al.* [34] also developed an ant colony optimization approach for MMKP. The authors have improved the convergence of their approach by integrating a conventional local search. The authors claim that their method is able to find solutions very good (close to optimal) in a relatively short computation time.
Htiouech *et al.* [31] [32] explore both sides of the feasibility border that consists in alternating both constructive and destructive phases in a strategic oscillating manner. In order to strengthen the surrogate constraint information, the authors enhance the method with constraints normalization. Numerical results show that the performance of his approach is competitive with previously published results.

This paper is organized as follows. Our heuristic for solving the MMKP using Lagrangian and surrogate relaxation informations are described in section 2. The computational results are reported in section 3. Section 4 summarizes the contributions of this work and discusses directions for further works.

II. Method in detail

In this section, we use a relaxation of the MMKP to define a choice rules in order to determine which item to add, drop or swap. This is accomplished by using respectively the Lagrangian [50] and surrogate relaxation [54][55] of the MMKP.

A. Foundation and basis

Glover and Kochenberger [16] introduced a critical-event tabu search approach which assumes that the memory structure is arranged around the feasibility border of the MDKP. This heuristic (referred to GK) uses a strategic oscillation that navigates both sides of the border to achieve a balance between intensification and diversification procedures. A parameter *span* is used to indicate the depth of the oscillation about the boundary, measured in terms of the number of variables added after crossing the boundary from the feasible side in a constructive phase and the number of variables dropped after crossing the boundary from the infeasible side in a destructive phase. Starting by a minimum value, the span is gradually increased to a maximum value. A series of constructive and destructive phases is performed for each value of the span parameter. When the span reaches the maximum value, it is gradually decreased to the minimum value. Once the span decreases to the minimum value, it is again gradually increased to the maximum value, and this oscillation process continues [16]. Hanafi and Freville [17] also demonstrated special version of this method that balances the interaction between intensification and diversification strategies for the MDKP. Tabu search fundamentals and strategies are widely discussed in Glover and Laguna [53]. Htiouech *et al.* [31][32] propose an adaptation of the oscillation strategy proposed by Glover and Kochenberger [16] to solve the MMKP.

B. Constructive phase

The add move is the main stage of the constructive phase of our oscillation approach. Add an object j of a group i to a solution S corresponding to the decision variable x_{ij} (initially equal to 0) to the current solution S^*, necessarily involve assignment to 1. In the classical oscillation strategy (like described in [16][17]), the constructive phase is divided into two sub-phases, namely feasible construction phase and unfeasible construction phase. In the case of MMKP, constructive phase can run only if the solution is infeasible. Note that in the case where the solution is feasible, any addition of a new item involves the immediate cross of the boundary of feasibility (choice constraint violated). Where the solution is infeasible, two scenarios are possible :

- there is one (or more) group(s) not containing any selected object in the current solution. In this case, the constructive phase (through its additions movements) is moving closer to the feasible region, since the number of groups without assigned objects decreases with each new insertion.
- each group i of n contains at least one item j selected. In this case, the constructive phase (through its additions movements) will increasingly distant from the border of feasibility navigating in the infeasible space. Indeed, each new insertion will cause a violation of the constraints of the problem more particular constraints of choice.

C. Destructive phase

The drop move is the main stage of the destructive phase of our oscillation approach. Remove an object from a solution, involve the allocation of the corresponding decision variable x_{ij} (initially equal to 1) to 0.

In the classical oscillation strategy, the destructive phase is divided into two sub-phases, namely feasible destructive phase and unfeasible destructive phase. In the case of MMKP, destructive phase occurs if the solution is infeasible. Note that in the case where the solution is feasible, any drop of an item from the current solution involves immediate cross of the boundary of feasibility (constraint violated choice). Where the solution is impractical, two scenarios are possible :

1) there are groups containing more than one selected item. In this case, the act of dropping items guide the research to get closer to the border of feasibility in the direction of the feasible space.
2) there are still groups containing no items selected in the current solution. The act of dropping an item from this solution distant the process of finding the feasible space. Indeed, each new deletion will cause a violation of more constraints of the problem.

D. Lagrangian relaxation based choice rules

The principle of Lagrangian relaxation is to dualise constraints of the original problem by introducing them multiplied by Lagrange multipliers in the objective function [41]. In other words, this relaxation technique is based to remove the most difficult constraints, and integrate them into the objective function by introducing weighting coefficients attempting to influence research in respect to the maximum possible constraints released. The new problem will be easier to solve, and its objective function will take into account the constraints relaxed. This relaxation has been proven as an effective tool for solving integer problems (Fisher [42]). Lagrangian relaxation provides a good upper bound for the MMKP (Moser *et al.* [43]). For more details, the reader may reference to [41][42]. In our approach, we relax the capacity constraints (2). Let the vector $\lambda \in \mathbb{R}^m$ (called Lagrange multipliers vector). The Lagrangian constraint relaxation associated to the resource constraints (2) is written :

$$LK(\lambda)\left[\begin{array}{ll} \mathit{Max}\Big\{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(c_{ij}-\sum_{k=1}^{m}\lambda^k a_{ij}^k\right)x_{ij}+\sum_{k=1}^{m}\lambda^k b^k\Big\} & \\ \quad \mathit{s.c.} \quad (3) & \\ \qquad\quad\;\; (4) & \end{array}\right. \tag{6}$$

The resulting problem is easy to solve, and the optimal solution x^* is given by :

$$x_{ij}^* = \begin{cases} 1 & \text{if } j = j^* \\ 0 & \text{else.} \end{cases} \tag{7}$$

with :

$$j^* = argmax\Big\{c_{ij} - \sum_{k=1}^{m}\lambda^k a_{ij}^k\Big\} \tag{8}$$

for $i = 1,\ldots,n$ et $j = 1,\ldots,n_i$. Since the second term $\sum_{k=1}^{m}\lambda^k b^k$ is a constant. Thus, the optimal value of the Lagrangian relaxation $L(\lambda)$ is written:

$$v(L(\lambda)) = \sum_{k=1}^{m}\lambda^k b^k + \left(\sum_{i=1}^{n}\text{Max}\Big\{c_{ij} - \sum_{k=1}^{m}\lambda^k a_{ij}^k\Big\}\right)^+ \tag{9}$$

for $j = 1,\ldots,n_i$. where $(\alpha)^+$=$max(0,\alpha)$. Knowing that each vector λ provides an upper bound, the best value is obtained by solving the Lagrangian dual expressed by:

$$Minv(L(\lambda)), \lambda \in \mathbb{R}^k. \tag{10}$$

The components of the vector λ are determined by three cases adapted heuristic H_{GK} Glover and Kochenberg [16], which depend on the state of the solution (feasible or not) and the direction of search.

We define the difference Δ^k as the remaining amount of the k resource after consumption needs of the current solution. In all cases, the quantity Δ^k is written:

$$\Delta^k = b^k - \sum_{i=1}^{n}\sum_{\{j/x_{ij}=1\}} a_{ij}^k \text{ or : } \Delta^k = b^k - \sum_{i=1}^{n}\sum_{j=1}^{n_i} a_{ij}^k x_{ij} \tag{11}$$

for all $k = 1, \ldots, m$.

Note that if $\exists k = 1, \ldots, m$ such that $\Delta^k < 0$ then k constraint is violated, and the solution is not feasible. The vector λ is determined by the following cases :

- Case 1 : if the solution is feasible (from constructive or destructive phase), the value λ^k corresponding to the k constraint is always equal to $\frac{1}{\Delta^k}$.
- Case 2 : when the solution is not feasible and research is in the constructive phase, we define Δ^k for all $k = 1, \ldots, m$ as follows:

$$\lambda^k = \begin{cases} \frac{1}{\Delta^k} & \text{if } \Delta^k > 0, \\ 2 + |\Delta^k| & \text{if } \Delta^k \leqslant 0. \end{cases}$$

- Case 3 : if the solution is not feasible and research is in the destructive phase, for all $k = 1, \ldots, m$, the multiplier λ^k is given by :

$$\lambda^k = \begin{cases} 0 & \text{if } \Delta^k \geqslant 0, \\ |\Delta^k| + \frac{1}{\sum_{i=1}^{n} \sum_{\{j/x_{ij}=0\}}^{n_i} a_{ij}^k} & \text{if } \Delta^k < 0. \end{cases}$$

In the case where $\Delta^k < 0$, λ^k can be simplified to :

$$\lambda^k = \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij}^k - b^k}$$

The choice rule for the constructive phase selects the variable x_{ij} to switch from 0 to 1 in order to

$$\mathit{Max} \left\{ r_{ij}^1 = (c_{ij} - \sum_{k=1}^{m} \lambda^k a_{ij}^k) | x_{ij} = 0 \right\} \tag{12}$$

The choice rule for the destructive phase of our approach selects the variable x_{ij} to switch from 1 to 0 in order to

$$\mathit{Min} \left\{ r_{ij}^1 = (c_{ij} - \sum_{k=1}^{m} \lambda^k a_{ij}^k) | x_{ij} = 1 \right\} \tag{13}$$

When the solution is feasible, swap moves are chosen to improve the quality of the current solution. We do this by selecting the variable x_{ij} to switch from 0 to 1. Otherwise we select the variable x_{ih} to switch from 1 to 0. The feature considered in our choice of x_{ij} and x_{ih} have to

$$\mathit{Max} \left\{ r_{ijh}^1 = (c_{ij} - c_{ih} - \sum_{k=1}^{m} (a_{ij}^k - a_{ih}^k) | x_{ij} = 0, x_{ih} = 1 \right\} \tag{14}$$

Where items j and h are in the same group i.

E. Add, drop and swap moves

It is attempting to note that in all what below r is r_1 when delay with Lagrangian relaxation, and it refers to r_2 when we use surrogate relaxation [32].

1) Add move: The add move is the principal move of the constructive phase of our approach. Adding a variable to the current solution S (initially equal to 0) is equivalent to set it up to 1. In this current step, the object added maximizes the quantity r_{ij} as described in algorithm 1.

Algorithm 1 procedure addMove(S)

$G^* = \mathit{Argmin} \left\{ \sum_{j=1}^{n_i} x_{ij} | i \in G \right\}$
$r_{i^*j^*} = \mathit{max} \left\{ r_{ij} | x_{ij} = 0, i \in G^*, j = 1, \ldots, n_i \right\}$
$x_{i^*j^*} = 1$
$S^* = S^* + x_{i^*j^*}$

2) Drop move: The heuristic gradually chooses which variables to drop during the destructive phase. Drop an object from the current solution S (initially equal to 1) is equivalent to reset it to 0. In this step, the dropped object minimizes the quantity r_{ij} as described in algorithm 2.

Algorithm 2 procedure DropMove(S)

$G^* = \mathit{Argmax} \left\{ \sum_{j=1}^{n_i} x_{ij} | i \in G \right\}$
$r_{i^*j^*} = \mathit{min} \left\{ r_{ij} | x_{ij} = 0, i \in G^*, j = 1, \ldots, n_i \right\}$
$x_{i^*j^*} = 0$
$S^* = S^* - x_{i^*j^*}$

3) Swap move: When the solution is feasible, our approach improves, step by step, its quality. The improvement should respect the feasibility of the solution and it is done by swap moves. Algorithm 3 shows how the objects to be swapped from the same group maximizes the quantity r_{ijh}.

Algorithm 3 procedure SwapMove(S)

$r_{i^*j^*h^*} = \mathit{max} \left\{ r_{ijh} | i \in G, j = 1, \ldots, n_i, h = 1, \ldots, n_i, \right\}$
such as $x_{ij} = 0, x_{ih} = 1$ and
$\sum_{i=1}^{n} \sum_{j=1}^{n_i} a_{ij}^k x_{ij} - a_{i^*h^*}^k + a_{i^*j^*}^k \leq b^k, k = 1, \ldots, m,$
and $c_{i^*h^*} < c_{i^*j^*}$
$x_{i^*j^*} = 1$
$x_{i^*h^*} = 0$
$S^* = S^* + x_{i^*j^*} - x_{i^*h^*}$

During the research, our approach can find several feasible solutions generated from the constructive or destructive phase of the research. Indeed, when this phase reaches its limits, the algorithm remembers only the best solutions enhanced with which he pursues research. The basic idea of this oscillation version is to intensify even more promising area by storing solutions found during the search intensification phase. We thus obtaining a list of feasible solutions (probably pretty good qualities) that we will use later as solutions departures for intensification (multi -start) the search area. This step is repeated until a number of iterations is reached without improvements .

III. EXPERIMENTATIONS AND TESTS

A. Experimental design

For our experimentations, we use the same benchmarks used by Hemendez and Dimopoulos [13]. The available resources, coefficients , were decreased by a factor f. First, no decrement is performed ($f = 1$); then, a decrement of 10% ($f = 0.9$) is allowed; finally, a decrement is done so that at least one of the

heuristics fails in finding a feasible solution. Because of the decrement factor used, a total of 90 problems were evaluated. The results are given in Tables I, II and III. In this section, the performance of Osc_{LAgr} and Osc_{surr} are determined on all instances of the literature given by Hemendez Dimopoulos and [13], offering themselves compare their approaches with those of Moser [43] and Khan *et al.* [44]. The first five columns of Tables I-III denote the number of the instance, the number of groups of objects, the number of objects in each group, the dimension of the vector capacity b^k, and finally the reduction factor f. We present the sixth column the results of Moser [43] followed by those of Khan *et al.* [44] (Heu), Hemendez and Dimopoulos [13] (Hmmkp). We provide in the last two columns owed results Osc_{LAgr} and Osc_{surr}.
We set up our approach parameters as follow :

- Tabu list size t = 4
- Initial solution: every variable is 0 ($x_{ij} = 0$ for all $i = 1, ..., n$ and $j = 1, ..., n_i$)
- *NumIter*= $n \times n_i$: number of iterations. Iteration corresponds to a pass of both a constructive phase and a destructive phase.
- *IterAutorized*=50 : number of iterations authorized without amelioration

B. Experimental results

To better visualize the improvements in quality of solutions we calculate the deviation value $\%dev$ given by the following formula:

$$\%dev = \left(1 - \frac{Osc}{Best}\right) \times 100 \qquad (15)$$

where $Best$ is the best value found. Found results shows that in relatively simple instances our approaches seems to be equivalent to literature ones. But, it is significantly clear to us that in the case of difficult instances our approaches efficiency is well proved. For this reason, and for more clarity we present, in our tables, only these difficult instances.

The values in bold indicate that our results are greater than or equal to the best results. We initially noted in the tables II-IV that our approaches oscillations could generate a feasible solution for all instances. A top view also shows that the quality of solutions is interesting. Greater than or equal to the best solution among those Moser, Heu and Hmmkp solutions are found in 72/90 cases, 80% of all instances. Improvements are particularly noted for the 30 most difficult instances (with the factor f is greater).

While algorithm Osc_{lagr} has made improvements on the order of 3.33%, 2.4% and 2.2% from Hmmkp mcknap7 respectively for instances, and mcknap8 mcknap9 Osc_{surr} has improved respectively 3.8%, 2.2% and 2.4%.

The Moser's method was unable to find a feasible solution for 30 instances, and for Heu, who could not find a feasible solution for 27 of the 90 instances.

Note that the rate of improvement (%Dev) becomes more significant for the last ten instances (larger reduction in resources) considered to be the most difficult to resolve. The rate of improvement has dyed 13.4% (for instance Mknapcb7-6-0.84), while occasionally the algorithm produces results of lower quality than Best (worst case a percentage of 1.77% to Mknapcb7-4-0.9). For instances of medium size (Mcknap8), the improvement rate reached 12.32% (for instance Mknapcb8-6-0.8), while the rate of degradation is the worst of 0.23% (Mknapcb8-7-0.9). Finally, an observation on the behavior of the algorithm for instances of larger sizes (mcknap9) allows us to note that we get the best solutions for almost last ten instances (the most difficult) and reached a rate improvement of 11.64% (instance Mknapcb9-3-0.75), with a worst case degradation of 0.08% (for Mknapcb9-7-0.9). As we already mentioned, the best improvement solutions are recorded for the hardest instances. Indeed, the first 20 instances of each file have the largest (including the first 10) available for solving instances resources. And their resolution does not require much effort and research converges quickly to the same solutions in 25/30 case Osc_{surr} (83%) and 23/30 cases Osc_{lagr} (i.e 76.6%). These results are explained by the fact that instances are not strongly correlated, and resources for the first 20 instances are relatively widely available. The research process and selects objects with a value selection criteria very distinguished in relation to other objects, where the immediate convergence to the same areas of research. The last 10 instances are obtained by reducing the resources available for the highest value of the factor f. Problems generated will therefore values criteria much tighter selection and then give more meaning to the memory frequency and recency to influence research to more promising areas. This is explained by the fact that the mechanisms of oscillations strategy (among others those of tabu search) will have no effect if whatever selection criteria is defined converges to the same selections object. We believe that our approaches differ especially if instances are relatively large size and if the ratings of the selection criteria are similar. Thus the most important improvements are recorded on average for the 10 most difficult instances and improves Osc_{lagr} Best 7.2%, 6.9% and 7.3% respectively for Mknapcb7, Mknapcb8 and Mknapcb9. Similarly, the same bodies Osc_{surr} improves a rate of 7.8%, 6.6%, and 7.9%. The results obtained show that the solutions of Osc_{surr} improve those obtained by Osc_{lagr} by an average rate of 0.3%.

IV. CONCLUSION AND FURTHER WORKS

In this paper, two new heuristics for the multichoice multidimensional knapsack problem Osc_{lagr} and Osc_{surr} are presented. Firstly, we introduce a new oscillation approach which explores both sides of the feasibility border to solve MMKP. Surrogate constraint information and Lagrangian relaxation are used to build the choice rules. Based on the computational results, better solutions than at other heuristics in the literature. Our approaches would be a very good candidate for time-critical applications such as adaptive multimedia systems where a near-optimal solution is acceptable, and fast computation is more important than guaranteeing the truly optimal value. Further works could be worthy, specially, we

mknapcb7	Parameters				Profiles solutions				
	n	n_i	m	f	Moser	Heu	Hmmkp	Osc_{lagr}	Osc_{surr}
0	20	5	30	0.9	15183	17560	17510	**18410**	**18459**
1	20	5	30	0.9	16439	17335	17948	17693	17827
2	20	5	30	0.9	16229	16907	17049	**17503**	**17503**
3	20	5	30	0.9	15901	17341	17883	17518	**17935**
4	20	5	30	0.9	16724	18127	17315	**18483**	**18483**
5	20	5	30	0.9	17754	17824	18318	**18561**	**18489**
6	20	5	30	0.9	15774	17307	18045	17883	17886
7	20	5	30	0.9	16301	17117	17301	**18122**	**18122**
8	20	5	30	0.9	16583	18213	17563	**18633**	**18633**
9	20	5	30	0.9	16146	16511	16691	**16995**	**17062**
0	20	5	30	0.84	*	*	15617	**16842**	**17052**
1	20	5	30	0.84	*	15885	15951	**16728**	**16671**
2	20	5	30	0.84	*	*	14080	**15059**	**15059**
3	20	5	30	0.84	*	*	14876	**16708**	**16712**
4	20	5	30	0.84	*	*	15595	**17028**	**17041**
5	20	5	30	0.84	*	*	15791	**17520**	**17399**
6	20	5	30	0.84	*	*	15484	**16968**	**16968**
7	20	5	30	0.84	*	*	14963	**16342**	**16342**
8	20	5	30	0.84	*	*	16160	**17285**	**17571**
9	20	5	30	0.84	*	15437	13098	**15889**	**16207**

* no feasible solution found

TABLE I
MKNAPCB7 : PERFORMANCE COMPARISON OF MOSER, HEU, HMMKP, Osc_{lagr} AND Osc_{surr}

mknapcb8	Parameters				Profiles solutions				
	n	n_i	m	f	Moser	Heu	Hmmkp	Osc_{lagr}	Osc_{surr}
0	50	5	30	0.9	44267	45535	45493	**45937**	**45937**
1	50	5	30	0.9	45963	47095	47130	**47166**	**47166**
2	50	5	30	0.9	42167	43810	45390	**45589**	**45789**
3	50	5	30	0.9	44732	45707	45810	45804	45804
4	50	5	30	0.9	43997	44669	45270	45170	45170
5	50	5	30	0.9	46579	46447	46611	46579	46579
6	50	5	30	0.9	45743	46224	46064	**46261**	**46261**
7	50	5	30	0.9	44038	44644	45450	45344	45387
8	50	5	30	0.9	46616	46939	47156	47138	47138
9	50	5	30	0.9	45195	45411	45859	**45930**	**45930**
0	50	5	30	0.8	*	41308	38523	**42227**	**42530**
1	50	5	30	0.8	*	*	41185	**44402**	**44143**
2	50	5	30	0.8	*	*	41259	**42747**	**42638**
3	50	5	30	0.8	*	*	40066	**42185**	**41975**
4	50	5	30	0.8	*	*	38262	**42458**	**42579**
5	50	5	30	0.8	*	*	39670	**42818**	**42164**
6	50	5	30	0.8	*	*	38547	**43295**	**42704**
7	50	5	30	0.8	*	*	39445	**41278**	**41411**
8	50	5	30	0.8	*	*	40954	**43775**	**43565**
9	50	5	30	0.8	*	40677	39834	**42435**	**42018**

* no feasible solution found

TABLE II
MKNAPCB8 : PERFORMANCE COMPARISON OF MOSER, HEU, HMMKP, Osc_{lagr} AND Osc_{surr}

mknapcb9	Parameters				Profiles solutions				
	n	n_i	m	f	Moser	Heu	Hmmkp	Osc_{lagr}	Osc_{surr}
0	100	5	30	0.9	90602	91879	92021	92004	92004
1	100	5	30	0.9	92371	92371	92371	**92371**	**92371**
2	100	5	30	0.9	93396	93294	93396	93367	93367
3	100	5	30	0.9	90137	91716	91815	91800	91800
4	100	5	30	0.9	92682	93150	93317	93257	93257
5	100	5	30	0.9	91002	91319	91547	91487	91487
6	100	5	30	0.9	90927	91322	91480	91430	91430
7	100	5	30	0.9	90861	91284	91672	91602	91602
8	100	5	30	0.9	93149	93034	93149	**93149**	**93149**
9	100	5	30	0.9	92562	93385	93528	93466	93466
0	100	5	30	0.75	*	*	74927	**81196**	**81835**
1	100	5	30	0.75	*	*	73570	**78053**	**78762**
2	100	5	30	0.75	*	*	74739	**79690**	**80846**
3	100	5	30	0.75	*	*	69813	**77940**	**78284**
4	100	5	30	0.75	*	*	74323	**80572**	**81587**
5	100	5	30	0.75	*	*	74303	**81097**	**81154**
6	100	5	30	0.75	*	*	72018	**77097**	**76766**
7	100	5	30	0.75	*	*	73777	**78184**	**77484**
8	100	5	30	0.75	*	*	74376	**78662**	**80180**
9	100	5	30	0.75	*	*	73496	**80818**	**80888**

* no feasible solution found

TABLE III
MKNAPCB9 : PERFORMANCE COMPARISON OF MOSER, HEU, HMMKP, Osc_{lagr} AND Osc_{surr}

propose to apply our approaches in the telecommunications problems like in [56].

REFERENCES

[1] C. Basnet and J. Wilson. Heuristics for determining the number of warehouses for storing non compatible products. International Transactions in Operational Research, 12(5):527-538, 2005.

[2] Y. Feng. Resource allocation in ka-band satellite systems. MSc Thesis, Institute for Systems Research, University of Maryland, 2001.

[3] A. Lardeux, G. Knippel, and G. Geffard. Efficient algorithms for solving the 2-layered network design problem. In Proceedings of the International Network Optimization Conference, Evry, France, pages 367-372, 2003.

[4] V. Li, G. Curry, and E. Boyd. Towards the real time solution of strike force asset allocation problems. Computers and Operations Research, 31(2):273-291, 2004.

[5] G. Raidl. The multiple container packing problem: A genetic algorithm approach with weighted codings. ACM SIGAPP Applied Computing Review, ACM Press, 7(2):22-31, 1999.

[6] J. Romaine. Solving the multidimensional multiple knapsack problem with packing constraints using tabu search. research report AFIT/GOR/ENS/99M-15, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 1999.

[7] Khan S, Quality adaptation in a multisession multimedia system : Model, algorithms and architecture, Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Victoria, Canada (1998).

[8] Khan S, Li KF, Manning EG, Akbar MM. Solving the knapsack problem for adaptive multimedia system. Studia informatica 2:161-82. (2002)

[9] Lee C, Lehoczky J, Rajkumar R, Siewiorek D. On quality of service optimization with discrete QoS options. In: RTAS'99: Proceedings of the fifth IEEE real-time technology and applications symposium. Washington, DC, USA: IEEE Computer Society; 1999. p. 276.

[10] Sadid MWH, Islam MR, Hasan SMK. A new strategy for solving multiple- choice multiple-dimension knapsack problem in pram model. In: Asian Applied Computing Conference. 2005.

[11] L. Chen, The utility model applied to layer-coded sources, Dept. of Computer Science, University of Victoria, Victoria, BC, Canada, (1998).

[12] Kellerer H, Pferschy U, Pisinger D. Knapsack problems. Berlin: Springer; 2004.

[13] Parra-Hemendez R, N.Dimopoulos. A New Heuristic for Solving the Multi-choice Multidimensional knapsack problem. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions. Volume: 35 , Issue: 5 Page(s): 708-717 (2005).

[14] Romain Picot-Clmente, Florence Mendes, Christophe Cruz, Christophe Nicolle. TOURISM-KM, A variant of MMKP applied to the tourism domain. ICORES 2012, Portugal (2012).

[15] Healy Jr. WC. Multiple choice programming. Operations Research;12:122 38. (1964)

[16] Glover F, GA.Kochenberger. "Critical event tabu search for multidimensional knapsack problems". In: Osman IH, Kelly JP, editors. Metaheuristics: theory and applications. p. 407-27. (1996)

[17] Hanafi S, Freville A. An efficient tabu search approach for the 0-1 multidimensional knapsack problem. European Journal of Operational Research ;106(23):659 75. (1998).

[18] Chu PC, Beasley JE. A genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics;4: 6386. (1998)

[19] Bertsimas D, Demir R. An approximate dynamic programming approach to multidimensional knapsack problems. Management Science;48(4):55065 (2002)

[20] SINHA, A. and ZOLTNERS, A. (1979) The multiple-choice knapsack problem. Oper. Res., 27, 503-515.

[21] R. Armstrong, D. Kung, P. Sinha, and A. Zoltners. A computational study of a multiple-choice knapsack algorithm. ACM Transactions on Mathematical Software, 9(2) :184-198, 1983.

[22] PISINGER, D. (1995) A minimal algorithm for the multiple-choice knapsack problem. Eur. J. Oper. Res., 83, 394-410.

[23] M. Akbar, O. Ergun, and A. Punnen. Heuristic solutions for the multiple-choice multi-dimensional knapsack problem. In International Conference on Computer Science, San Francisco, USA, 2001.

[24] Y. Toyoda. A simplified algorithm for obtaining approximate solutions

to zero-one programming problems. Management Science, 21(12) :1417-1427,1975.

[25] H. Pirkul. Efficient algorithms for the capacitated concentrator location problem. Computers and Operations Research, 14(3) :197-208, 1987.

[26] M. Akbar, M. Rahman, M. Kaykobad, E. Manning, and G. Shoja. Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls. Computers and Operations Research, 33(5) :1259-1273, 2006.

[27] C. Hiremath and R. Hill. New greedy heuristics for the multiple-choice multi-dimensional knapsack problem. International Journal of Operational Research, 2(4) :495-512, 2007.

[28] Ilog. Ilog CPLEX 10.0 Reference Manual. 9, rue de Verdun, BP 85, F-92453, Gentilly, France, 2006.

[29] Hanafi. S, R. Mansi, and C. Wilbaut. Iterative relaxation-based heuristics for the multiple-choice multidimensional knapsack problem. volume 5818 of Lecture Notes in Computer Science, pages 73-83. Springer Verlag, 2009.

[30] Hanafi. S, Christophe Wilbaut : Improved convergent heuristics for the 0-1 multidimensional knapsack problem. Annals OR 183(1) : 125-142 (2011)

[31] Skander Htiouech, Sadok Bouamama and Rabeh Attia. Article: OSC: solving the multidimensional multi-choice knapsack problem with tight strategic Oscillation using Surrogate Constraints. International Journal of Computer Applications 73(13):1-7, July 2013. Published by Foundation of Computer Science, New York, USA.

[32] Skander Htiouech, Sadok Bouamama, Rabeh Attia: Using surrogate information to solve the multidimensional multi-choice knapsack problem. IEEE Congress on Evolutionary Computation 2013: 2102-2107

[33] I. Crvits, S. Hanafi, R. Mansi, and C. Wilbaut. Iterative semi-continuous relaxation heuristics for the multiple-choice multidimensional knapsack problem. Computers and Operations Research Volume 39, Issue 1, 2012, Pages 32-41

[34] S. Iqbal, M. Bari, and M. Rahman. Solving the multi-dimensional multi-choice knapsack problem with the help of ants. In Proceedings of the 7th international conference on Swarm intelligence (ANTS 2010), 2010.

[35] Z. Ren and Z. Feng. An ant colony optimization approach to the multiple-choice multidimensional knapsack problem. In Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO 2010). Association for Computing Machinery, 2010.

[36] Mhand Hifi, Slim Sadfi, Abdelkader Sbihi. "An exact algorithm for the multiple-choice multidimensional knapsack problem". Cahiers de la MSE b04024, Maison des Sciences Economiques, Universit Paris Panthon-Sorbonne. (2004).

[37] A. Sbihi. A best first search exact algorithm for the multiple-choice multidimensional knapsack problem. Journal of Combinatorial Optimization, 13(4):337-351, 2007.

[38] M. Razzazi and T. Ghasemi. An exact algorithm for the multiple-choice multidimensional knapsack based on the core. Advances in Computer Science and Engineering, Communications in Computer and Information Science, 6:275-282, 2008.

[39] Hifi. M, M. Michrafy, and A. Sbihi. Heuristic algorithms for the multiple-choice multidimensional knapsack problem. Journal of the Operational Research Society, 55(12):1323-1332, 2004.

[40] T. Ghasemi and M. Razzazi. Development of core to solve the multidimensional multiple-choice knapsack problem. Computers and Operations Research, 60:349-360, 2010.

[41] Geo-rion AM. Lagrangian relaxation for integer programming. Mathematical Programming Study 1974;2:82-114.

[42] M. L. Fischer. The Lagrangean relaxation method for solving integer programming problems. Management Sciences, 27 :1-18, 1981.

[43] Moser M, Declarative scheduling for optimally graceful QoS degradation, in Proc. IEEE Int. Conf. Multimedia Computing Systems (ICMCS), pp. 86 93(1996).

[44] Khan S, Quality adaptation in a multisession multimedia system : Model, algorithms and architecture, Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Victoria, Canada (1998).

[45] Dyer ME, Walker J. Dominance in multi-dimensional multiple-choice knapsack problems. Asia Pacific Journal of Operational Research;15(2):15968. (1998)

[46] Mhand Hifi, Slim Sadfi, Abdelkader Sbihi. "An exact algorithm for the multiple-choice multidimensional knapsack problem". Cahiers de la MSE b04024, Maison des Sciences Economiques, University paris pantheon-Sorbonne. (2004)

[47] Khan S, Li KF, Manning EG, Akbar MM. Solving the knapsack problem for adaptive multimedia system. Studia informatica 2:161-82. (2002)

[48] Hifi, M., Michrafy, M., Sbihi, A.: Heuristic algorithms for the multiple-choice multidimensional knapsack problem. Journal of the Operational Research Society 55,

[49] Watson R, The optimal admission and adaptation of service level agreements in packet networks: Applying the utility model, Masters thesis, Dept. of Electrical and Computer

[50] Geo-rion AM. Lagrangian relaxation for integer programming. Mathematical Programming Study 2:82114. (1974)

[51] Cherfi M. Hifi, A column generation method for the multiple-choice multi-dimensional knapsack problem, Comput. Optim. App. online first, Springer Netherlands (2008).

[52] Hifi M, M. Michrafy and A. Sbihi, "A reactive local search-based algorithm for the multiple-choice multi-dimensional knapsack problem", Computational Optimization and Applications, vol. 33, 271-285, (2006)

[53] Glover F, Laguna M. Tabu search. Boston: Kluwer Academic Publishers; (1998).

[54] Glover F. Surrogate constraints: tutorial notes. October (1998).

[55] Glover F. Heuristics for integer programming using surrogate constraints. Decision Sciences; 8:15666. (1977)

[56] Athanasios S. Lioumpas, Petros S. Bithas, Angeliki Alexiou: Partitioning of Distributed MIMO Systems based on Overhead Considerations. CoRR abs/1211.3719 (2012)