Theoretical Informatics and Applications Will be set by the publisher
Informatique Théorique et Applications

A NOTE ON A TWO DIMENSIONAL KNAPSACK PROBLEM WITH UNLOADING CONSTRAINTS *

Jefferson Luiz Moisés da Silveira[1], Eduardo Candido Xavier[1] and Flávio Keidi Miyazawa[1]

Abstract. In this paper we address the two-dimensional knapsack problem with unloading constraints: we have a bin B, and a list L of n rectangular items, each item with a class value in $\{1, \ldots, C\}$. The problem is to pack a subset of L into B, maximizing the total profit of packed items, where the packing must satisfy the unloading constraint: while removing one item a, items with higher class values can not block a. We present a $(4+\epsilon)$-approximation algorithm when the bin is a square. We also present $(3+\epsilon)$-approximation algorithms for two special cases of this problem.

1991 Mathematics Subject Classification. 68W25,05B40,90C27.

1. Introduction

In this paper we study the two-dimensional knapsack problem with unloading constraints (KU), that is a generalization of the well known NP-Hard two-dimensional knapsack problem [1, 13]. This problem arises in operations research transportation problems, like the 2L-CVRP (Capacitated Vehicle Routing Problem with Two Dimensional Loading Constraints Problem) [6, 10]. In this problem we are given k identical vehicles, with a weight capacity and a rectangular surface that may be accessed only from one side. We are also given a graph that represents the customers, the distance between them and the starting point of the vehicles (a special vertex on the graph which represents the depot). Each customer has a demand given by a set of rectangular items. The goal is to find k routes that

Keywords and phrases: Knapsack Problem, Approximation Algorithms, Unloading/loading Constraints

* *This research was supported by CNPQ and FAPESP.*

[1] Institute of Computing, University of Campinas - UNICAMP, Av. Albert Einstein, 1251, Campinas, Brazil; e-mail: jmoises@ic.unicamp.br & ecx@ic.unicamp.br & fkm@ic.unicamp.br

visits all clients such that the total cost of the routes is minimized. For each route it must be obtained a feasible packing of the items of the clients in the route: the unloading of items of a client must not be blocked by items of customers to be visited later along the route (unloading constraint).

The KU problem can be formally defined as follows: We are given a bin B of width and height 1, and n items of C different classes, each item a_i with height $h(a_i)$, width $w(a_i)$, profit $p(a_i)$ and class $c(a_i)$. A packing is feasible if items do not *overlap*, all of them are packed inside bin B, and there is an *order* to unload the items, such that no item of a given class blocks the way out of other items of smaller classes. We consider that while removing one item of B only horizontal movements are allowed. The class values c represents the order in which items must be removed. While removing one item, only this item can be moved and only in the available free space of the bin. In this case, if an item a_i is packed in (x_i, y_i) (i.e. its bottom left corner is placed at this position) and a_j is packed in (x_j, y_j) and $c(a_j) > c(a_i)$ then either $y_j + h(a_j) \leq y_i$ or $y_i + h(a_i) \leq y_j$ or $x_j + w(a_j) \leq x_i$. These constraints guarantee that item a_j is not blocking a_i during its removal while using only horizontal movements (see Fig. 1).

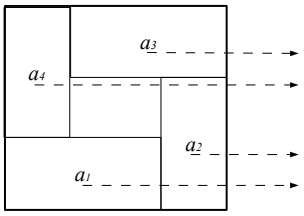

FIGURE 1. Suppose we have a squared bin B and four items $a_i, i = 1, 2, 3, 4$ with $c(a_2) = c(a_3) = 1$ and $c(a_1) = c(a_4) = 2$. Any 2-staged feasible solution for this problem uses at most 3 items, while the pattern above uses all 4 items and is feasible for the KU problem. If we had $c(a_2) > c(a_1)$ then this packing would be infeasible since a_2 would be blocking a_1.

Since items are removed in non-decreasing order of values c, we can assume that, in a feasible packing, when removing item a_i, only items a_j with class $c(a_j) \geq c(a_i)$ are still packed. The other items a_k with $c(a_k) < c(a_i)$ should be removed previously. The value of a feasible solution is the sum of the profits of items packed in B.

The KU problem is a generalization of the classical two dimensional Knapsack Problem: a special case in which all items have the same class value c.

Denote by $OPT(I)$ the cost of an optimal packing for the instance I and $A(I)$ the cost of the solution computed by algorithm A. The proposed algorithms have polynomial-time complexity and satisfy $sup_I \frac{A(I)}{OPT(I)} \leq \alpha$, where α is the approximation ratio.

Related Work. An extended abstract of this work appeared in [3] and to our knowledge, there are no approximation algorithms for the KU problem. There are a few heuristics focused on the 2L-CVPR Problem [5,6,10,16], and some heuristics for the strip packing version of the problem [4]. There is also an exact algorithm for the three dimensional version of the KU problem, based on an ILP formulation which take into account other practical constraints within the unloading constraint [14, 15]. Furthermore, the recent advances in the two dimensional knapsack does not directly apply to the KU problem. In 2006, Harren [7] proposed a $(5/4 + \epsilon)$-approximation algorithm for the knapsack problem where the items are squares and have arbitrary profits and the bin is also a square. Jansen and Solis-Oba, proposed a PTAS for packing squares with independent profits into a *rectangle* [12]. There is a $(2 + \epsilon)$-approximation algorithm for packing rectangles with arbitrary profits [13]. In [11], Jansen and Prädel proposed a PTAS for the case in which the profits are proportional to the items surface. These results cannot be directly used in the KU problem because of the unloading constraints. It is worth noting that 2-staged knapsack solutions clearly satisfies the unloading constraints, but approximation algorithms for such problems do not lead directly to approximation algorithms for the KU problem. In [11], for instance, it is presented a PTAS for the geometrical 2-staged knapsack problem. But their result does not imply a PTAS to our problem, since optimal solutions for the KU problem may not satisfy the cutting pattern required in the 2-staged knapsack problem (See Fig. 1). Thus the approximation results for 2-staged problems can not be directly used.

Main Results. In this paper we present the first approximation algorithms for the KU problem. Our algorithm combines algorithms for two-dimensional knapsack problems and level based algorithms for the bin packing problem. We design approximation algorithms for 3 variations of the KU problem: a $(3 + \epsilon)$-approximation algorithm for the case where the goal is to maximize the profit of squares packed into a square; a $(4 + \epsilon)$-approximation algorithm for the general case where the goal is to maximize the profit of rectangles packed into a square; and a $(3 + \epsilon)$-approximation algorithm for the geometrical case where the profits corresponds to the area of the rectangles. We also show some results for the case in which orthogonal rotations are allowed.

The paper is organized as follows. In Section 2 we present a general result that leads to three approximation algorithms. In Section 3 we present an $(4 + \epsilon)$-approximation algorithm for the general case. Finally, in Section 4 we draw some conclusions.

2. A Hybrid Algorithm for the KU problem

In this Section we present a general method to generate approximation algorithms for several versions of the KU problem. Using it we present approximation algorithms for three variations of the problem.

We define a *regular* two-dimensional knapsack problem, as a two-dimensional Knapsack problem in which:

- The items can not overlap.
- The items and the bin have rectangular form;
- Any level based packing is a valid (not necessarily optimal) solution to the problem;

For instance the classical Geometrical and Guillotined cases of the two-dimensional Knapsack problem are *regular*.

First we will prove a general result: given an algorithm for a *regular* two-dimensional knapsack problem and another level based algorithm for the two-dimensional bin packing problem, we can construct another algorithm for the same knapsack problem with the unload constraint.

By level based algorithm we mean an algorithm that packs items into levels (shelves), where items in some level are "separated" from items in other levels: all items in a level $l_0 = 0$ are packed with their bottom at the bottom of the bin B, and the items of a subsequent level l_{i+1} are packed with their bottom in a line above all the items at level l_i such that $l_{i+1} = l_i + \max_{a_k \in l_i}(h(a_k))$. Notice that items in the same level can be arranged horizontally in an arbitrary manner without breaking the levels structure. The algorithms *First-Fit Decreasing Height* (FFDH) and *Next-Fit Decreasing Height* (NFDH) are examples of level based algorithms for the Strip packing problem [2] and the algorithm *Hybrid First-Fit* (HFF) is a level based algorithm for the two-dimensional bin packing problem [8].

Denote by $p(B_i)$ the sum of the profits of items packed into the bin B_i, and denote by $p(L)$ the sum of the profits of items in the list L.

Theorem 2.1. *Let $\mathcal{A}_K$ be an α-approximation algorithm for a regular 2D knapsack problem. Let $\mathcal{A}_{BP}$ be an absolute β-approximation level based algorithm for the 2D bin packing problem. Then there is an $(\alpha\beta)$-approximation algorithm, denoted by $\mathcal{A}_{KU}$ for the same version of the knapsack problem and that respects the unloading constraints.*

Proof. The Algorithm $\mathcal{A}_{KU}$ is presented in Algorithm 1. It uses algorithm $\mathcal{A}_K$ (an α-approximation algorithm for a *regular* 2D knapsack problem) and algorithm $\mathcal{A}_{BP}$ (an absolute β-approximation algorithm for the 2D bin packing problem that is based on levels).

In line 3 of Algorithm 1, algorithm $\mathcal{A}_K$ selects a list of items $L' \subseteq L$ as a solution for the knapsack problem. Then in line 4 of Algorithm 1, the level based algorithm $\mathcal{A}_{BP}$ generates the packing P of the items in L' into bins. Finally, the Algorithm $\mathcal{A}_{KU}$ selects the bin with the largest total profit among the created bins, sort each level in this bin by class and return it as a solution (lines 5-8 of Algorithm 1).

Algorithm 1 $\mathcal{A}_{KU}$ algorithm

1: **Input:** Algorithms $\mathcal{A}_K$ and $\mathcal{A}_{BP}$ and a list L of items of C different classes.
2: **Begin**
3: $L' \leftarrow \mathcal{A}_K(L)$;
4: $P \leftarrow \mathcal{A}_{BP}(L')$;
5: Let $B_1, \ldots, B_m$ be the bins in P.
6: Let B be the bin with largest total profit $p(B)$ among the bins in P.
7: Sort items in each level of B by non-increasing order of class.
8: return B.
9: **end**

First notice that the items in B are partitioned into levels since it is generated by a level based algorithm $\mathcal{A}_{BP}$. The items in each level of B are sorted by non-increasing order of class. This guarantees that the final packing satisfies the unloading constraint. Also, since the problem is *regular* then the items can be packed in levels.

Now consider the total profit in bin B. Denote by OPT_K the value of an optimal packing for the *regular* 2D knapsack problem, and OPT_{KU} the value of an optimal packing for the *regular* 2D knapsack version of the problem with unloading constraints. Given a list of items L, we have $OPT_K(L) \geq OPT_{KU}(L)$.

Assume that $\mathcal{A}_{BP}(L')$ returned m bins. Since $\mathcal{A}_{BP}$ is an absolute β-approximation algorithm, we have $m \leq \beta$. Let $(B_1, \ldots, B_m)$ be the set of bins returned by the algorithm $\mathcal{A}_{BP}$. We have

$$\begin{aligned} mp(B) &\geq \sum_{j=1}^{m} p(B_j) \\ &= p(L') \\ &\geq \frac{1}{\alpha} OPT_K(L) \\ &\geq \frac{1}{\alpha} OPT_{KU}(L). \end{aligned}$$

Therefore,

$$p(B) \geq \frac{1}{m\alpha} OPT_{KU}(L) \geq \frac{1}{\alpha\beta} OPT_{KU}(L).$$

From the previous inequality, we conclude that $\mathcal{A}_{KU}$ is an $(\alpha\beta)$-approximation algorithm for the 2D knapsack problem with unloading constraints.

□

In [8], van Stee and Harren proved that the HFF algorithm, for the two-dimensional bin packing problem, is an absolute 3-approximation when bins are squares and items cannot be rotated, and is an absolute 2-approximation when 90

degree rotations are allowed. So, using the HFF algorithm, which is a level based algorithm, we can prove the following results applying Theorem 2.1:

- Let $\mathcal{A}_K$ be the PTAS proposed in [12] for the 2D knapsack problem where items are squares, with profits, and the bin is also a square. Then there is a $(3 + \epsilon)$-approximation algorithm for the same version of the problem and that respects the unloading constraints. If rotations are allowed then the algorithm is a $(2 + \epsilon)$-approximation.
- Let $\mathcal{A}_K$ be the $(2 + \epsilon)$-approximation algorithm proposed in [13] for the 2D knapsack problem, where items are rectangles with profits, and the bin is a square. Then there is a $(6 + \epsilon)$-approximation algorithm for the same 2D knapsack problem with unloading constraints. If rotations are allowed then the algorithm is a $(4 + \epsilon)$-approximation.
- Let $\mathcal{A}_K$ be the PTAS proposed in [11] for the 2D knapsack problem where items are rectangles with profits that are proportional to their area (this problem is known as Geometrical Knapsack), and the bin is a square. Then there is a $(3 + \epsilon)$-approximation algorithm for the 2D Geometrical Knapsack problem with unloading constraints. If rotations are allowed then the algorithm is a $(2 + \epsilon)$-approximation.

Let $n = |L|$ and $T_{\mathcal{A}}(n)$ be the time complexity of algorithm $\mathcal{A}$. It is important to notice that the time complexity of the algorithm $\mathcal{A}_{KU}$ is bounded by $O(T_{\mathcal{A}_K}(n) + T_{\mathcal{A}_{BP}}(n) + O(n \log n))$, with $T_{\mathcal{A}_K}(n)$ from line 3 on Algorithm 1, $T_{\mathcal{A}_{BP}}(n)$ from line 4 on Algorithm 1 and $O(n \log n)$ from line 7 on Algorithm 1. Thus, assuming $\mathcal{A}_K$ and $\mathcal{A}_{KU}$ being polynomial-time algorithms, then so is $\mathcal{A}_{KU}$.

3. A 4-approximation Algorithm for the KU Problem

In this section we present a $(4+\epsilon)$-approximation algorithm for the general 2D KU problem, without rotations, where items have arbitrary profits and the bin is a square. This result improves the $(6+\epsilon)$-approximation algorithm of the previous section.

Let $\mathcal{A}_\epsilon$ be the $(1+\epsilon)$-approximation algorithm for the one-dimensional knapsack problem with arbitrary profits [9], and consider the NFDH algorithm [2], which is a level based algorithm for the strip packing problem. The NFDH starts by sorting the items by non-increasing order of height. Then, starting from the bottom of the strip as its first level, the algorithm packs items in order on the current level (at the bottom of the level and left justified) as long as they fit; then the current level is closed and the same procedure is applied to a new level above the current level while there are items to be packed.

In Algorithm 2 we present our $\mathcal{A}'_{KU}$ algorithm for the KU Problem. It starts selecting a set L' of items with the algorithm $\mathcal{A}_\epsilon$ where each 2D item a_i of the original instance is transformed into an item for the 1D knapsack problem: the size of the item, $s(a_i)$, has value equal to the area of a_i and the profit values are the same (line 3-4 of Algorithm 2). Then it uses the NFDH algorithm to generate a strip packing with the 2D items in L' (lines 5 of Algorithm 2). Finally the

algorithm packs the levels generated by NFDH into 4 bins, and selects the one with the largest total profit as a final solution (lines 6-8 of Algorithm 2).

Algorithm 2 $\mathcal{A}'_{KU}$ algorithm

1: **Input:** A list L of 2D items of C different classes.
2: **Begin**
3: Let L_u be the list L of items where each item now has size $s(a_i) = h(a_i) \cdot w(a_i)$, and profits values remain the same.
4: Let L' be the original 2D items selected by algorithm $A_\epsilon(L_u)$.
5: Generate the packing P with NFDH(L').
6: Pack the levels of P into at most 4 bins.
7: Let B be the bin with largest total profit $p(B)$ among the 4 created bins.
8: Sort each level in B by non-increasing order of class.
9: return B.
10: **end**

Theorem 3.1. *The $\mathcal{A}'_{KU}$ algorithm is a $(4+\epsilon)$-approximation algorithm for the KU problem.*

Proof. Denote by $Area(L)$ the total area of items in a given list L. First, notice that the algorithm A_ϵ selects L' such that $Area(L') \leq 1$ and $p(L') \geq \frac{1}{1+\epsilon} OPT_K(L)$ where $OPT_K(L)$ is the value of an optimal solution for the 2D knapsack problem.

Since NFDH(L') $\leq 2Area(L') + 1$ (see [2]), we have that NFDH(L') ≤ 3.

Let P be the strip packing solution computed by NFDH algorithm. We claim that there are at most three levels with height larger than $1/2$ in this packing (the height of some level is the height of the highest item packed on it). The levels are sorted by height in P. Suppose for the purpose of contradiction that the first four levels have height $> 1/2$. Then all items packed in the first, second and third levels have height $> 1/2$. The first item packed in the fourth level also has height $> 1/2$. It is easy to see that the items of the first and second level have total area $> 1/2$ and the items of the third level together with the first item of the fourth level also have total area $> 1/2$. But this contradicts the fact that $Area(L') \leq 1$, and so at most three levels have height larger than $1/2$.

Now we show how to pack all levels of P into at most 4 bins. Pack each level with height larger than $1/2$ in three different bins B_1, B_2 and B_3 (if there are less than 3 levels with height $> 1/2$ consider the 3 largest levels). Now for the remaining levels pack them in the first bin until for the first time the total height is larger than 1, then proceed to the second bin, and finally to the third. Notice that only the first and second bins may have total height larger than 1, since the total height of all levels is at most 3. Pack the last packed levels of bin B_1 and B_2 in another bin B_4 (this can be done since each one of these levels have height $< 1/2$).

Finally the algorithm selects the bin B with largest total profit among B_1, B_2, B_3 and B_4. The items in the levels of this bin are sorted in non-increasing order of

class values. This guarantees that we have a feasible packing for the KU problem. Finally we have

$$\begin{aligned} 4 \cdot p(B) &\geq \sum_{i=1}^{4} p(B_i) \\ &= p(L') \\ &\geq \frac{1}{1+\epsilon} OPT_K(L) \\ &\geq \frac{1}{1+\epsilon} OPT_{KU}(L). \end{aligned}$$

That is,

$$p(B) \geq \frac{1}{4+\epsilon'} OPT_{KU}(L)$$

Therefore, the algorithm $\mathcal{A}_{KU}$ is a $(4+\epsilon)$-approximation algorithm for the KU problem.

□

Moreover, the analysis of Theorem 3.1 is tight. Consider the following instance with $|L| = 7$ in the form $(h(a_i), w(a_i), p(a_i), c(a_i))$:

$$\{(1, 2\varepsilon, \alpha, 1), (\frac{1}{2}+2\varepsilon, 1-\varepsilon, \beta, 1), (\frac{1}{2}+\varepsilon, 2\varepsilon, \alpha, 1), (\frac{1}{4}+\varepsilon, 1-\varepsilon, \beta, 1), (\frac{1}{4}, 2\varepsilon, \alpha, 1), (\frac{1}{8}+\varepsilon, 1-\varepsilon, \beta, 1), (\frac{1}{8}, 2\varepsilon, \alpha, 1)\}.$$

We can choose ε small enough such that $Area(L) < 1$. We also choose a large value for α and a small value for β. In line 4 of the algorithm $\mathcal{A}'_{KU}$, we will have $L' = L$. Furthermore, the NFDH algorithm just pile the items one above the other since it sorts the items in non-increasing order of height and uses next fit to pack items into levels.

Finally, the algorithm $\mathcal{A}'_{KU}$ is going to use 4 bins to pack the list L' (See Fig. 2).

After packing all the items into the bins, the $\mathcal{A}'_{KU}$ algorithm chooses the 2^{nd} bin with profit $\alpha + 2\beta$. An optimum solution packs all thin items (the ones with width 2ϵ) into one bin, side by side with value 4α. Choosing $\alpha \to \infty$ and $\beta \to 0$ we have

$$\frac{OPT_{KU}(L)}{\mathcal{A}'_{KU}(L)} = \lim_{\alpha\to\infty, \beta\to 0} \frac{4\alpha}{\alpha + 2\beta} = 4.$$

Finally, let $n = |L|$ and $T_{A_\epsilon}(n)$ be the time complexity of algorithm A_ϵ. The time complexity of the algorithm $\mathcal{A}'_{KU}$ is bounded by $O(T_{A_\epsilon}(n) + O(n \log n))$, with $T_{A_\epsilon}(n)$ from line 4 on Algorithm 2 and $O(n \log n)$ from lines 5 and 8 on Algorithm 2. Thus, assuming A_ϵ being a polynomial-time algorithm, so is $\mathcal{A}'_{KU}$.

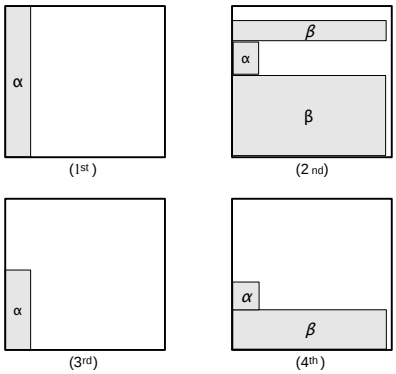

FIGURE 2. The algorithm uses the 4^{th} bin since two items did not fit into bins 1 and 2.

4. Concluding Remarks

In this paper we consider some variants of the 2D knapsack problem with unloading constraints. To our knowledge, this is the first paper to present approximation results to this problem. We presented a $(6+\epsilon)$-approximation algorithm for the general case of the problem, and two $(3+\epsilon)$-approximation algorithms, one for the special case where the profits are proportional to the areas of the items and another one for the version where items are squares. Finally, we improve our first result for the general case and present a $(4+\epsilon)$-approximation algorithm and proved that this result is tight.

References

[1] A. Caprara and M. Monaci. On the two-dimensional knapsack problem. *Operations Research Letters*, 32:5 – 14, 2004.

[2] E. G. Coffman Jr., M. R. Garey, D. S. Johnson, and R. E. Tarjan. Performance bounds for level-oriented two dimensional packing algorithms. *SIAM Journal on Computing*, 9(4):808–826, 1980.

[3] J. L. S. da Silveira, E. C. Xavier, and F. K. Miyazawa. Two dimensional knapsack with unloading constraints. In *VI Latin American Algorithms, Graphs and Optimization Symposium (LAGOS 2011)*, Electronic Notes in Discrete Mathematics, pages 1–4. 2011.

[4] Jefferson L. M. da Silveira, Flávio Keidi Miyazawa, and Eduardo C. Xavier. Heuristics for the strip packing problem with unloading constraints. *Computers & OR*, 40(4):991–1003, 2013.

[5] B. L. P. de Azevedo, P. H. Hokama, F. K. Miyazawa, and E. C. Xavier. A branch-and-cut approach for the vehicle routing problem with two-dimensional loading constraints. In *Simpósio Brasileiro de Pesquisa Operacional - SBPO*, Porto Seguro, Brazil, 2009.

[6] M. Gendreau, M. Iori, G. Laporte, and S. Martello. A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1):4–18, October 2007.

[7] R. Harren. Approximating the orthogonal knapsack problem for hypercubes. In *ICALP (1)*, volume 4051 of *Lecture Notes in Computer Science*, pages 238–249, 2006.

[8] R. Harren and R. van Stee. Absolute approximation ratios for packing rectangles into bins. *J. of Scheduling*, 2010.

[9] O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22:463–468, October 1975.

[10] M. Iori, J-J. S-González, and D. Vigo. An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41(2):253–264, May 2007.

[11] K. Jansen and L. Prädel. How to maximize the total area of rectangles packed into a rectangle. Technical Report 0908, Christian-Albrechts-Universität zu kiel, Italy, 2009.

[12] K. Jansen and R. Solis-Oba. A polynomial time approximation scheme for the square packing problem. In *Proc. of the Integer Programming and Combinatorial Optimization*, volume 5035 of *LNCS*, pages 184–198. Springer Berlin / Heidelberg, 2008.

[13] K. Jansen and G. Zhang. On rectangle packing: maximizing benefits. In *Proc. of the 15th ACM-SIAM Symposium on Discrete Algorithms*, pages 204–213, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[14] L. Junqueira, R. Morabito, and D. Yamashita. Abordagens para problemas de carregamento de contêiners com considerações de múltiplos destinos. *Gestão & Produção.*, 18(1), 2011.

[15] L. Junqueira, R. Morabito, and D. S. Yamashita. Mip-based approaches for the container loading problem with multi-drop constraints. *Annals OR*, 199(1):51–75, 2012.

[16] E. E. Zachariadis, C. T. Kiranoudis, and C. D. Tarantilis. A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195:729–743, 2009.

Communicated by (The editor will be set by the publisher).
May 29, 2013.