A Guided Tabu Search for the Vector Bin Packing Problem

Farida Mannai
InterVPNC laboratory
FSJEG,University of Jendouba
Jendouba,Tunisia
Email: farida.mannai@gmail.com

Mongi Boulehmi
InterVPNC laboratory
FSJEG,University of Jendouba
Jendouba,Tunisia
Email: mongiboulehmi@gmail.com

***Abstract*— The importance of BPP stems from the fact that it has vast industrial applications and it frequently occurs as a subproblem of various practical problems. The present work addresses Two-dimensional Bin Packing Problem. We evaluate the performance of a Tabu Search (TS) algorithm applied to this problem. In our work the Two-dimensional Bin Packing Problem is defined as follows: given a set of N elements of various weights ($w1,w2,w3,\ldots,w_n$) and of various volumes ($v1,v2,v3,\ldots,v_n$) these elements must be packed into a limited number of M bins having the same capacity C and the same volume V. The main objective of the BPP here is to minimize the number of used bins while respecting their capacity C and their volume V. In this paper the problem is NP-hard optimization problem which is not possible to find an efficient algorithm to optimally solve large size instances in reasonable computational time. To solve this problem, we adapt a meta-heuristic algorithm called the tabu search.**
We illustrate the proposed approach by a numerical experimentation and we analyze the generated results. For comparison, the instances were also solved using the CPLEX solver but as a result tabu search resulted relatively optimal solutions in a shorter time frame and it can also resolve ,in opposition to CPLEX, the NPHard problems.

Keywords— Two Dimensional Bin Packing, Tabu Search, CPLEX.

I INTRODUCTION

Combinatorial optimization is a discipline that is of major interest both theoretically and practically. On the theoretical side, many highly innovative concepts have been proposed during the last century. These concepts continue to be enriched by a rich and very dense bibliography. In practical terms, this discipline is experiencing a resurgence of major interest, particularly in terms of logistics, a discipline in which the optimization is essential.

The Bin-Packing problem also allows modeling many problems with assignment constraints and some scheduling problems. In all cases, very rich nature of this problem has given rise to several classifications in the literature. Now many variations of this problem appear in the literature, such as 2D packing, 3D packing, linear packing, packing by weight, packing by cost, and many others. They are used in several industrial applications, logistics, computer and even publishing field, such as filling up containers, loading trucks with weight capacity, creating file backups in removable media and loading ships and planes. This problem is an NP-hard since it generalizes the classical problem of bin-packing one dimensional (1BP) known as an NP-hard problem. Given that the Bin-Packing problem is NP-hard, the enumeration of all feasible solutions to find the best solution is impossible even for a medium size problem. However, one can approach the resolution by approximate methods, in particular heuristics and meta-heuristics.

In this work we adapt a meta-heuristic method (the Tabu Search) to solve this problem, this method will be referred to as method exact using CPLEX. As with many other metaheuristic used to resolve this problem, the success of tabu search is, in large part, due to its ability to steer the search process from getting stuck in a local optimum. TS is characterized by its flexibility which has a flexible memory that keeps the process from cycling in one neighborhood of the solution space. TS provides the ability to support multi-objective optimization. Therefore, it can cover large search space and to find its promising regions though it may take a relatively short time to reach a relatively optimal solution. It is known as non-population meta-heuristics (single initial solution), which is less complex and requires less computing time, than other approaches which are known as population meta-heuristics (Particle Swarm Optimization(PSO), Ant Colonies (AC) and Genetic Algorithms (GA)).

I-A Tabu search for Two Dimensional Bin Packing Problem

In this work, we shall present the tabu search technique for the two dimensional BPP. Its various ingredients have shown a remarkable efficiency on many problems. Many computational experiments have shown that tabu search has now become an established optimization technique which can compete with almost all known techniques and which by its flexibility can outcome many classical procedures [1],[2]. Among the optimization approaches and iterative techniques play an important role for most optimization problems no procedure is known in general for its ability to get directly an approximately optimal solution. The general step of an iterative procedure consists in constructing from a current solution i, a next solution j and in checking whether one should stop there or perform another step. Neighborhood search methods are iterative procedures in which a neighborhood N(i) is defined for each feasible solution i, and the next solution j is searched among the solutions in N(i). Tabu search can be considered as neighborhood search methods. The basic ingredients of tabu search are described in the next section.

I- The TS-BPP Algorithm

We detail in what follows the TS-BPP algorithm for the BPP. The algorithm starts with an initial solution generated by the first fit decreasing (FFD) procedure that initializes a single bin with items that are processed in the decreasing order. When there is no more space in the first bin to store the current item, a second bin is opened but without closing the first. When we have r bins opened and an item to store, the process starts from the first one, if all bins can't handle it, a new bin $(r+1)$ is then used without closing the other. Until the list of items becomes empty. For each bin r in the FFD solution, we compute the remaining space (Ar) [10]. To enhance the solution, we include it into an improvement process using TS-BPP. In order to improve the efficiency of the exploration process, one needs to keep track not only of local information (like the current value of the objective function) but also of some information related to the exploration process. This systematic use of memory is denoted tabu list L as an essential ingredient in our procedure.

As a first step toward the description of TS-BPP, we initialize an empty tabu list $L[i][j]$. In order to keep information on the itinerary through the last solutions visited, in addition to the value $f(s*)$ of the best solution $s*$ visited so far. Such information will be used to guide the move from s to the next solution s0 to be chosen in neighborhoods N(s).

The neighborhood generation is the next step in the TS-BPP process, Neighbors of a current solution described by a several moves obtained by swapping (exchanging) items between all possible pairs of bins [23]. We use 2 different swapping schemes: Swap $(1,0)$, Swap $(1,1)$, proposed by Fleszar and Hindi (2002).
In Swap $(1,0)$, we consider moving one item i from bin X to bin Y , while respecting the capacity c of bin. Which means that we must ensure the condition that the remaining space A[r] in bin r can handle the weight wi of the next item from bin $r+1$. And before finishing the move, we update the residual space in bin r and $r+1$. After this move, test if r+1 is empty (equal to C) then decrease the number of bins. In Swap $(1,1)$, we swap item i from bin X with item j from bin Y , while also respecting the capacity c of bins. To avoid the risk of visiting a solution again and omit the risk of cycling, we need to save the moves in the tabu list (a tabu move). This is the point where the use of memory (the tabu list L) is helpful to forbid moves which might lead to recently visited solutions.

II TABU SEARCH IMPLEMENTATION

This section describes the detailed implementation of the algorithm's starting solution, move definition and solution neighborhood, aspiration criteria and diversification and intensification strategies.

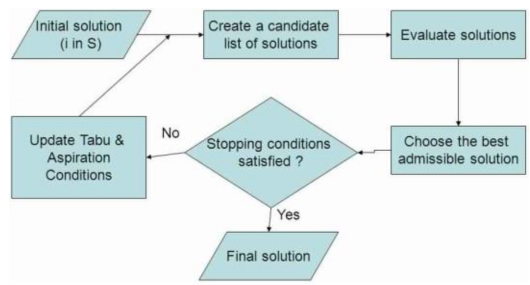

Figure 1: Flowchart of a Standard Tabu Search algorithm

Algorithm 1 The TS-BPP Algorithm

Let N be Non-increasing list of elements;
Invoked FFD heuristics for initial solution S° *;*
Take S° *like input for the TS procedure;*
Initial empty tabu list $L[i][j]$*;*
for i := 0 *to number of items do;*
for j := 1 *to number of items-1 do;*
Definition of possible "moves" to generate neighborhoods N(S)' of S° *;*
IF $A[r]$ *"the remaining space in bin r " can pack the next item from bin r +* 1*;*
Make the Swaps $(1-0)$*;*
Save the move in the tabu list $L[i][j] = 1$ *;*
Test IF r + 1 *is empty (equal to C) then decrease* S° *;*
Apply Transfers $(1-1)$ *interchanging of two elements ;*
Save the move in the tabu list ;
end for ;
end for;
The process ends when number of iteration is reach ;
Or IF we get the best solution ;

II-A Initial solution

This kind of algorithm starts with an initial solution which is modified during execution in order to find a near optimal solution. The initial solution can be generated at random or can be generated by a function. In this section we describe and analyze a simple heuristic used to initialize our tabu search algorithm.
The heuristic is based on a technique called the First Fit (FF). We try to fit item as soon as possible by scanning the list of bins in increasing time of open order to find the first suitable bin and fit into it. It requires $\theta(n \log n)$ time, where n is the number of elements to be packed.

The algorithm can be outlined as follows.

Algorithm 2 Algorithm First Fit Decreasing for 2DBPP

```
for All objects i = 1,2,..., n do
for All bins j = 1,2,.. do
if Object i fits in bin j then
    Pack object i in bin j
    Break the loop and pack the next object.
  end if
  end for
    if Object i did not fit in any available bin then
    Create new bin and pack object i
  end if
end for
```

II-B Neighborhood exploration

Besides the calculation of an initial solution, the choice of the neighborhood is the main problem-specific part of a local search heuristic. This part has a very important influence on the efficiency of local search heuristics. It determines the way in which we navigate through the solution space to find the best neighbor or an improving neighbor.
We have to define a neighborhood structure on the set S of feasible solutions: we have to specify a corresponding set of neighbors $N(s)$ for each solution S. In principle, these sets N(s) may be arbitrary subsets of S. Furthermore, we need some systematic ways of defining neighborhoods since, otherwise, it is not possible to store the neighborhood.

The structure of the neighborhood for our tabu search heuristic focuses on changes of assignments of items to bins [8], [23]. It is a composite neighborhood which uses the swap operator. The classification depends on the history of the search, particularly as manifested in the recency or frequency that certain move or solution components, called attributes, have participated in generating past solutions. For example, one attribute of a swap is the identity of the pair of elements that change positions. As a basis for preventing the search from repeating swap combinations tried in the recent past, potentially reversing the effects of previous moves by interchanges that might return to previous positions, we will classify as tabu all swaps composed of any of the most recent pairs of such bins.

II-C Tabu list structure

The tabu list contains information about a certain number of previous iterations, the last T iterations. The tabu list records the last assignments of an item to a bin. This prevents the reversal of the assignment status as long as they remain in the list. Given a solution in the composite neighborhood, the tabu moves are built as follows: If we swap item i in bin A with item j in bin B, we forbid the assignment of i to A and j to B.
In general, a tabu is specified by some attributes of the moves. When a move having an attribute e is performed, a record is maintained for the reverse attribute e, in order to restrict a move having some subset of the reverse attributes.

II-D Stopping criteria

These are the conditions under which the search process will terminate. In this study the search will terminate if one of the following criteria is satisfied:

- The number of iterations since the last change of the best solution is greater than a prespecified number;

  Or
- The number of iterations reaches the maximum allowable number.

II-E Diversification and intensification strategies

The roles of intensification and diversification in tabu search are already implicit in several of the preceding prescriptions. Diversification strategies seek to generate solution S that embody compositions of attributes significantly different from those encountered previously during the search. A different way which guaranties the exploration of unvisited regions is to penalize frequently performed moves or solutions often visited.

For an intensification strategy, we choose S to be a small subset of elite solutions that share a large number of common attributes, and secondarily whose members can reach each other by relatively small numbers of moves. During the intensification phase the moves or the solutions are evaluated taking into account their amount of "good" components.

III EXPERIMENTAL DESIGN

The TS heuristic is implemented in Java language (NetBeans IDE 6.9) based on a 2 GHz Intel Core 2 Duo processor with 3 GO RAM under Windows 64 mode.

IV COMPARING THE PERFORMANCE OF METHOD EXACT AND TABU SEARCH IN 2D-BPP

CPLEX is, at the base, a solver of linear programs. It is marketed by the company ILOG. As such, it is based on a powerful implementation of the primal simplex. It also features dual simplex and network simplex. The problems addressed by the ILOG optimization suite are: mixed linear and linear programs, mixed quadratic and quadratic programs, quadratic stress and mixed quadratic stress programs. It is an optimizer for mathematical programming.

While experiencing it is possible to use CPLEX in an interactive by invoking a dedicated shell. We notice that this optimizer has its flaws and constrains. It is proved that it takes a long period of time into solving the problem specially dealing with large instances.

In this part, we are going to analyze the table I of solution of the exact method using CPLEX. Since we had no previous results to compare to, we used the CPLEX solver as a reference solution. CPLEX displayed some difficulties solving the test instances.

Through the process of solving the problem, we noticed first that the number of bins depends on the number of items. To clarify this point, in some cases whenever the number of bins is further less than the number of items, CPLEX does not result any solution, as in the cases number 3,7 and 13 in the table below.

We have modified only the number of the bins in the case number 14 we added three bins regarding the problem number 13 in the case, CPLEX generates a solution.

Second, taking the examples number 9,10 and 11 we guarded some number of bins and items but we have increased the volume V and capacity C in this case we note that when the volume and capacity are increasing the time of execution of CPLEX is decreasing.

Finally, whenever we increase the number of bins and items, the execution time of CPLEX increases also. As in the case number 16, CPLEX took 527,05 seconds to give us a solution to the problem of packing 35 items in 30 bins. On some of the instances CPLEX has imposed a time limit of 3 hours but was still unable to solve the instances for example case number 17 in our table.

Studying the table II below and experiencing the effectiveness of tabu search we have noticed that this heuristic solved in a minimum time frame all of the given cases. All the solution resulted by utilizing tabu search are too close to the exact solutions.

Instances					CPLEX				TS		
Cases	n items	n bins	C	V	Sol	Time(sec)	Iter	Gap	Sol	Time(s	Gap
1	10	9	100	100	8	0,06	36	5,75	8	0,1	5,75
2	10	9	1000	1000	8	0,06	46	5,75	8	0,1	5,75
3	10	5	1000	1000	-	0,06	4	-	4	0,1	2,45
4	15	9	100	100	-	0,06	43	-	6	0,1	3,76
5	10	10	100	100	10	0,06	59	6,40	10	0,1	6,40
6	15	13	100	100	10	0,06	68	6,40	11	0,2	5,40
7	20	10	100	100	-	0,06	47	-	6	0,2	2,31
8	20	15	100	100	14	0,33	8871	13,43	14	0,2	8,29
9	25	20	100	100	16	4,65	125038	6,25	17	0,2	7,25
10	25	20	1000	1000	20	1,08	21094	5,00	21	0,2	6,00
11	25	20	10000	10000	17	0,37	5962	9,46	18	0,3	9,46
12	30	25	100	100	18	1606,12	31867598	5,56	19	1,11	6,56
13	30	25	1000	1000	-	0,75	15605	-	25	1,10	0,00
14	30	27	1000	1000	26	0,81	14671	9,37	25	1,10	0,00
15	35	30	100	100	21	72,23	1223313	4,76	22	1,11	9,36
16	35	30	1000	1000	24	527,05	10750427	4,17	25	1,20	7,47
17	40	35	100	100		12373,47			29	1,23	5,46

Table 1: Comparison results on the 2DBPP test instances.

The figure 2 shows the performance of the TS-BPP for each class of instances relative to the Method Exact, at each point in the continuous curve we note the number of hits for the TS-BPP and their corresponding for the Method Exact presented in the discontinuous curve. The result after interpreting this figure shows that the TS-BPP outperforms the Method Exact in a reasonable time frame.

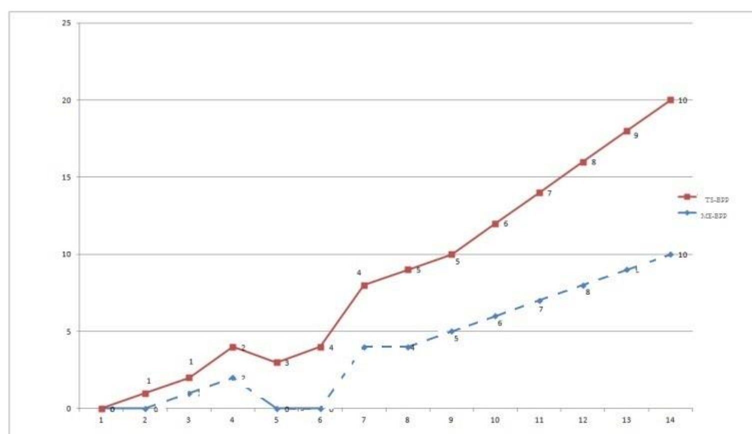

Figure 2: A graphical illustration on the performances of the

TS-BPP

V CONCLUSION

To sum up, in this paper we have studied the two dimensional bin packing problem using an unstudied previously couple of dimensions, that are the volume v and weight w. Tacking our objective, which is minimizing the number of bins utilized in a minimum time frame.

Solving this problem, we have chosen to compare the solutions resulted from two different algorithms, that are CPLEX, as method exact, and tabu search as the meta-heuristic method. Interpreting the performance of CPLEX and tabu search, we conclude that the meta-heuristic method is more effective than the exact method into solving the 2D-BPP.

Though, tabu search gave as a more approximative solution to the exact method with a highly effective performance, we may study other meta-heuristics and compare their performance to tabu search.

As we can result, tabu search's maximum time frame is $1{,}23$ seconds, in opposition to the maximum time frame given by method exact using CPLEX, which is $1606{,}12$ seconds concerning the studied cases in this paper. We conclude, that tabu search is faster than the CPLEX solver on some of the hardest instances. For many of the medium-sized instances, however, the CPLEX solver showed the best performance.

REFERENCES

[1] Stephen C.H.Leung , *Extended localsearchalgorithmbasedonnonlinearprogramming for twodimensionalirregularstrippackingproblem*, Computers& OperationsResearch39(2012).

[2] Lijun Wei, *A skyline heuristic for the 2D rectangular packing and strip packing problems*, European Journal of Operational Research 215 (2011).

[3] Defu Zhang , *A heuristicblock-loadingalgorithmbasedonmultilayersearchforthe container loadingproblem*, Computers & OperationsResearch39(2012).

[4] Christoforos Charalambous, *A constructivebinorientedheuristicforthetwo-dimensionalbinpacking problem withguillotinecuts*, Computers & OperationsResearch38(2011).

[5] Nikhil Bansal, *Two-dimensional bin packing with one-dimensional resource augmentation*, Discrete Optimization 4 (2007).

[6] Soo Y. Chang , *A two-dimensional vector packing model for the efficient use of coil cassettes*, Computers & Operations Research 32 (2005).

[7] Andrea Lodi, *Two-dimensional packing problems: A survey*, European Journal of Operational Research141 (2002).

[8] A. Lodi, S. Martello, and D. Vigo, *Neighborhood search algorithm for the guillotine non- oriented two-dimensional bin packing problem*, Kluwer Academic Publishers, Boston, 1998.

[9] A. Lodi, S. Martello, and D. Vigo, *Approximation algorithms for the oriented two- dimensional bin packing problem*, European Journal of Operational Research, 1999.

[10] A. Lodi, S. Martello, and D. Vigo, *Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems*, INFORMS Journal on Computing,1999.

[11] W. Schneider, *Trim-loss minimization in a crepe-rubber mill; optimal solution versus heuristic in the 2 (3)-dimensional case*, European Journal of Operational Research,1988.

[12] F. J. Vasko, F. E. Wolf, and K. L. Stott, *A practical solution to a fuzzy two-dimensional cutting stock problem*, Fuzzy Sets and Systems,1989.

[13] K. Lagus, I. Karanta, *Paginating the generalized newspaper: A comparison of simulated annealing and a heuristic method*, In Proceedings of the 5th International Conference on Parallel Problem Solving from Nature,Berlin,1996.

[14] B. E. Bengtsson, *Packing rectangular pieces a heuristic approach* , The Computer Journal, 1982.

[15] F. K. R. Chung, M. R. Garey, and D. S. Johnson, *On packing two-dimensional bins*, SIAM Journal of Algebraic and Discrete Methods,1982.

[16] J. O. Berkey and P. Y. Wang, *Two dimensional finite bin packing algorithms*, Journal of the Operational Research Society,1987.

[17] J. B. Frenk and G. G. Galambos, *Hybrid next-fit algorithm for the two-dimensional rectangle bin-packing problem* , Computing,1987.

[18] A. El-Bouri, N. Popplewell, S. Balakrishnan, and A. Alfa, *A search based heuristic for the two-dimensional bin-packing problem*, INFOR,1994.

[19] S. Martello and D. Vigo, *Exact solution of the two-dimensional finite bin packing problem*, Management Science, 1998.

[20] H. Dyckhoff and U. Finke, *Cutting and Packing in Production and Distribution*, Physica Verlag, Heidelberg, 1992.

[21] K. A. Dowsland and W. B. Dowsland, *Packing problems*, European Journal of Operational Research,1992.

[22] H. Dyckhoff, G. Scheithauer, and J. Terno, *Cutting and Packing* , Annotated Bibliographies in Combinatorial Optimization, 1997.

[23] Yan Jin, Jin-Kao Hao, *General swap-based multiple neighborhood tabu search for the maximum independent set problem*, 2014 Elsevier Ltd. All rights reserved.

=2