A Novel Model for Pallet Loading Problem Using Hybrid Optimization

Ruddy P. P. Cunha
MIA, Universidade de Fortaleza UNIFOR
Av. Washington Soares 1321, Fortaleza, CE, Brasil, 60811-905, Brasil
ruddypaz@unifor.edu.br

Alvaro M. S. Neto[1], Alexei B. Aguiar[2], Plácido R. Pinheiro[3], Andre Coelho[4]
MIA, Universidade de Fortaleza UNIFOR
Av. Washington Soares 1321, Fortaleza, CE, Brasil, 60811-905, Brasil
({netosobreira[1], alexei[2]}@edu.unifor.br), ({placido[3], acoelho[4]}@unifor.br)

ABSTRACT

A Pallet is a squared structure used for storage. As it has been produced over five million pallets a year, methodologies that reduce the storage cost are interesting. It is necessary to best allocate the items, which we've used as boxes, to take the best advantage of the storage space. There are many possible allocations for a box on the pallet, and simply solve the problem can take a long time. The usage of Hybrid Methodology has proved that time of problem solving has significantly been reduced. This methodology consists on the reduction of the possible allocations for the boxes. It modifies the original problem, producing new and smaller problems which have a solving time smaller than the original one. The characteristics from the original problem are preserved and its solution is one feasible original problem's solution. Some of them can be the optimal solution for the trouble. Executing many instances these small problems, can be found the optimal solution of the main problem.

KEYWORDS. Pallet, Hybrid Methodology, Combinational Optimization

1. Introduction

A pallet is a plain structure used to facilitate the transportation of the most variables goods, since it can be loaded several objects at the same time. Typically, a pallet is made by wood. However, there are pallets made by other materials like plastic and metal, and each type of pallet has its owns advantages and disadvantages. Most of the pallets support a load of about 1ton of weight. As it has many advantages in the load transportation, nowadays, is a produced over five hundred million pallet a year. According to Kocjan and Holmström (2008), when the number of items to be loaded on the pallet increases, the cost of transportation decreases, turning the pallet loading problem an economical problem. The cost reduction is not the only thing that changes. The complexity of the problem increases when the number of items increases too.

As Kocjan and Holmström (2008) say, the items to be loaded on the pallet are stored in boxes that must be placed with its edges parallel with the pallet's edges. The vertical orientation of the boxes is fixed. In this paper, it will be discussed that the pallet loading problem is a particular case of the cutting & packing problem. Another point to observe is the stability of the pallet. We have assumed that pallet will always be in its maximum capacity, and then it will be stable and will not be needed to worry about stability in the mathematical model. The pallet loading problem consists of load the maximum number of boxes on a pallet. The pallet has a given length and width. The type of box has a given length and width either. And the boxes must be loaded orthogonally on the pallet. The box's position can be shifted on its length and width, but not on the height.

Hybrid Methodology consists of extract smaller problems that have the same characteristics and its solution is one feasible solution of the original problem. As the problems extracted are smaller, their solving time are smaller either, they can be executed in several sub-problems and find a feasible solution in a time smaller than it would be if the original problem was executed. The Genetic Algorithm was used by this methodology to manage the sub-problems.

It is impracticable to try to allocate the box on each single coordinate of the pallet, so, it was created a discretization matrix. This matrix has the coordinates where the boxes can be effectively allocated. For example: the pallet has its dimension L x W (length L, width W) and the boxes have its dimensions l x w (length l, width w). If a box is allocated at the coordinate (x, y) of the pallet, the next coordinates that a box could possibly be allocated are the ones that are not in a place that the previously allocated box is. Doing this, the numbers of coordinates to be tested decreases, decreasing the solving time either. Not just this, the discretization allows the extraction of the sub-problems. By eliminating some coordinates of the discretization matrix, we can obtain a sub-problem of the original one. Using Genetic Algorithm to control which coordinates will be eliminated, it can be obtained a sub-problem that can give a feasible solution in a small time. Such time can be so small, that many different instances of sub-problems can be executed without overcoming the solving time of the original problem.

As said before, the Pallet Problem is a particular case of the Cutting & Packing Problem, and the height of the boxes can be used to decompose the problem of a bi-dimensional and one-dimensional. Consequently, the total height over the pallet can be despised either. The implementation of the model used is a bi-dimensional one. So it will be used a cutting and packing model for the pallet problem.

2. The Constrained Two-Dimensional Non-Guillotine Cutting Problem

The Constrained Two-dimensional Non-guillotine Cutting Problem consists of cutting rectangular pieces from a single large rectangular object. Each piece is of fixed orientation and must be cut with its edges parallel to the edges of the object. The number of pieces of each type that are cut must lie within prescribed limits and, in addition, the cuts may not go from one end to another. Each piece has an associated value and the objective is to maximize the total value of the pieces cut. This problem has been shown to be NP-Complete Dowsland and Dowsland (1992), meaning that it is impossible to find their optimal solution by resorting to an enumerative, brute-force approaches alone, except in trivial cases.

In Dowsland and Dowsland (1992), two combinatory methods that generate constrained cutting patterns by successive horizontal and vertical builds of ordered rectangles are investigated to tackle this problem. Each of the algorithms uses a parameter to bound the maximum waste they may create. Error bounds measure how close the pattern wastes are to the waste of the optimal solution. These algorithms are fast and can yield efficient solutions when applied to small problems. A tree search approach is presented based upon the lagrangean relaxation of a 0-1 integer linear programming formulation of the problem to derive an upper bound on the optimal solution. The formulation makes use of variables that relate to whether or not a piece of a particular type is cut with its bottom-left hand corner at a certain position.

In Fekete and Schepers (1997), the authors combined the use of a data structure for characterizing feasible packing´s with new classes of lower bounds, as well as other heuristics, in order to develop a two-level tree search algorithm for solving high-dimensional packing problems to optimality. In that approach, projections of cut pieces are made onto both the horizontal and vertical edges of the stock rectangle. Each such projection is translated into a graph, where the nodes in the graph are the cut pieces and an edge joins two nodes if the projections of the corresponding cut pieces overlap. They show that a cutting pattern is feasible if and only if the projection graphs have certain properties. The authors have shown that problems of considerable size can be solved to optimality in reasonable time with this approach.

3. Outline of Some Hybrid Methodology Approaches

Over the last years, interest in hybrid metaheuristics has risen considerably among researchers in combinatorial optimization. Combinations of methods such as simulated annealing, tabu search and evolutionary algorithms have yielded very powerful search methods Talbi (2002). This can be evidenced by the diversity of works about this topic found in the literature. In Martin et al.(1992), for instance, it is introduced a hybrid approach that combines simulated annealing with local search heuristics to solve the traveling salesman problem. In Mahfoud and Goldberg (1995), it is presented a hybrid method that applies simulated annealing to improve the population obtained by a genetic algorithm. In Chu (1997), a local search algorithm, which utilizes problem-specific knowledge, is incorporated into the genetic operators of a GA instance to solve the multi-constraint knapsack problem.

In this regard, it has become ever more evident that a skilled combination of concepts stemming from different metaheuristics can be a very promising strategy one should resort to when having to deal with complicated optimization tasks. The hybridization of metaheuristics with other operations research techniques has been shown great appeal as well, as they typically

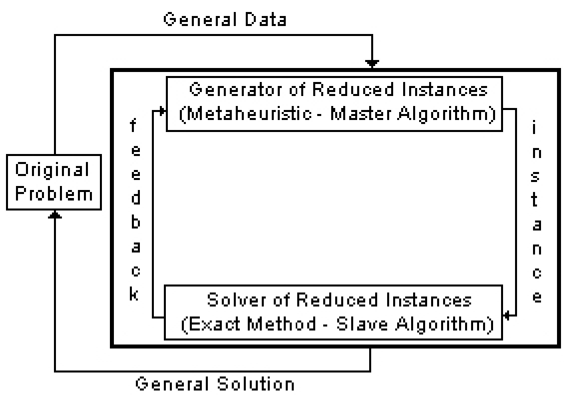

Figure 1: The hybrid framework under investigation

represent complementary perspectives over the problem solving process as a whole. In general, combinations of components coming from different metaheuristics and/or from more conventional exact methods into a unique optimization framework have been referred to by the label of "hybrid metaheuristics" by Talbi (2002), Raidl (2006).

In this context, a hybrid methodology has been recently introduced in the literature by Nepomuceno et al. (2006), Nepomuceno et al. (2007a), Nepomuceno et al. (2007b) and Nepomuceno et al. (2008), trying to push forward the boundaries that limit the application of an exact method through the decomposition of the original problem into two conceptual levels. According to the framework underlying this approximative methodology (see Figure1) the exact method (encapsulated in the Solver of Reduced Instances (SRI) component) works no more with the original problem but with reduced instances (i.e. sub-problems) of it that still preserve its conceptual structure. By this means, an optimal solution to a given subproblem will also be a feasible solution to the original problem. On the other hand, the metaheuristics component of the framework works on a complementary optimization problem, that is, the design of reduced instances of the original problem formulated as mathematical programming (viz., integer linear programming (ILP) models. It is referred to as the Generator of Reduced Instances (GRI), whose goal is to determine the subset of points of the reducible structure that could derive the best subproblem instance; that is, the sub-problem which, when submitted to the SRI, would bring about the feasible solution with the highest possible objective function value. In this scenario, the objective function values of the solutions that could be realized by the solver are used as figure of merit (fitness) of their associated sub-problems, thus guiding the metaheuristic search process. The interaction between GRI and SRI is iterative and repeats until a given stopping condition is satisfied.

It is only rather recently that hybrid algorithms which take ideas from both exact and heuristic local search techniques have been proposed. These techniques are traditionally seen as pertaining to two distinct branches of research toward the effective solution of combinatorial optimization problems, each one having particular advantages and disadvantages. Therefore, it appears to be a straightforward idea to try to combine these two distinct techniques into more powerful algorithms Gehring et al.(1990). There have been very different attempts to combine strategies and methods from these two scientific streams. Some of these hybrids mainly aim at providing optimal solutions in shorter time, while others primarily focus on getting better heuristic solutions Puchinger and Raidl (2005). For instance, in order to reduce the search space, in Vasquez and Hao (2001), it is combined tabu search with an exact method for solving the 0-1 multidimensional knapsack problem. By other means, in Cook and Seymour(2003), it is proposed a two-phase hybrid method where high quality tours for the traveling salesman problem are generated, and the sub-problem induced by the set of previous tours is solved exactly on the restricted graph.

In Talbi(2002), the author has recently presented a taxonomy of hybrid metaheuristic components, which distinguishes the hybridization into two levels. In the low-level scheme, the result is a functional composition of a single optimization method. In this hybrid class, a given function of a metaheuristic is replaced by another metaheuristic. Conversely, in the high-level scheme, the different metaheuristics are self contained, and there is no direct relationship to the internal workings of the others. By other means, in Puchinger and Raidl(2005), an alternative classification of existing approaches combining exact and metaheuristic algorithms for combinatorial optimization is presented, which distinguishing the following two main categories: (i) collaborative combinations; and (ii) integrative combinations. By collaboration, it means that the constituent algorithms exchange information to each other, but none is part of the other. The algorithms may be executed sequentially, intertwined, or in parallel. By integration, it means that one technique is a subordinate component of the other. Thus, there is the distinction between a master algorithm – which can be either an exact or a metaheuristic algorithm – and a slave algorithm.

4. Parameters of Genetic Algorithm

4.1 Initial Population Generation

One of the aspects that asked for amelioration concerns the generation of the initial population and maintenance of diversity along the first generations. For instance, no care was taken in avoiding the creation of repeated individuals in the first rounds. Also, no control over the distribution of gene alleles was performed, in a manner as to avoid the possibility of genetic drift, i.e., the premature loss of some important gene values. The methodology relied exclusively on the mutation operator to grant the needed diversity of alleles.

To address these problems, the initial population generation is now based on a fixed uniform probability distribution over the gene alleles controlled by the IDD parameter. Such parameter determines the approximated mean density of alleles in the initial generation. We mean as "density of an individual" the ratio between the number of its genes having '1' as allele and the size of the chromosome. Equation 1 formally defines the density d(x) of an individual x that has n as total number of genes. This parameter is then calibrated manually (a not so hard task) before the framework execution. When the density rate is very low, it may happen that some individuals have an empty gene group associated to the x and y axis discretization sets, taking a pallet problem as reference. These individuals would then generate subproblems with infeasible solutions. When each individual is generated, a validity checking mechanism is in charge of detecting this kind of anomaly and then execute a fixing-up routine that sets the first allele of the empty discretization set as '1'. At this point, the newly-created individual is ensured to be a valid one. Yet the phenomenon of allele extinction (genetic drift) can still occur, although this can be controlled by setting higher values to the initial density rate.

$$x = \{a_i | i = 1 \ldots n\}, d(x) = \frac{\sum_{i=1}^{n} a_i}{n} \quad (1)$$

4.2 Density Control Operator

Another negative factor that has limited the effectiveness of the original methodology is the uncontrolled density explosion. The increase in density tends to generate subproblems closer to the original problem yielding better solutions. This aspect can be pictured as an attractor pushing the population density up as the GA evolves. Nevertheless, this behavior has an undesirable side effect. Usually, high densities imply higher complexity to be dealt with by the mixed integer-linear method, indirectly affecting the search process con- ducted by the metaheuristic. This trend leads the evolutionary process of the genetic algorithm to a progressive increase of the mean population density. Thus at each generation the evolutionary process becomes slower until its exhaustion. This causes a drastic limitation in the number of genetic algorithm generations and also reduces its effectiveness.

Another parameter incorporated into the methodology is the ideal density IDD, which is also implemented by a routine and calibrated beforehand. During the density control operation, the following scheme is conducted over each individual. If the individual density is lower than IDD, nothing is changed. On the other hand, if the density is higher than the ideal value, some of the individual's genes having null assigned values are randomly chosen to be deactivated (i.e., zeroed). The density that exceeds the threshold IDD triggers a density reduction algorithm. After this stage, if the current density value is still greater than the ideal density, the other genes with some associated credit will also suffer deactivation, those with smaller credits being deactivated first. This last procedure is optional and its execution in controlled by the parameter PNZCG.

4.3 Gene Rotation Rate

There is another characteristic that acts as a limiting factor. Many individuals may have a waste of volume in the range between the axis origin and the first gene that has its allele equals to

1. The chromosomes have groups of genes associated with the axis x and y For instance, let's consider a chromosome c. Its two gene groups are A and B associated to the x and y axis respectively. For each gene groups we may consider the first gene with its allele equals to 1. We can define these particular genes as:

$$g_x \mid x = \min\{i \mid A_i = 1\} \quad (2)$$
$$g_y \mid y = \min\{j \mid B_j = 1\} \quad (3)$$

So the wasted volume w caused by this factor on the solution generated from c on a pallet with length l and width w is:

$$w = lwh - (l - pos(g_x))(w - pos(g_y)) \quad (4)$$

where pos(g) returns the axis position associated to the gene g. This problem arises from the free genes allocation natures that have no direct and strict compromise with the allocation on the beginning of the axis on the coordinate (0, 0). The relation is indirect so after many GA generations some high fitness individuals may have learned that this position optimizes the total box allocation area but many others may not have acquired this behavior. It is better suited if all individuals have this waste eliminated during its entire lifetime.

4.4 Population Size – PS

Controls how many individuals the GA population will have since there is no population size variation mechanism in this implementation. This parameter is first used in the initial population generation. After that, all operators will be influenced because they have to deal with all individuals of the population. There is a tradeoff on this parameter. When the population size is increased there will be more diversity on the function landscape exploration. This maximizes the probability of finding good optimum points. The evil effect of this population size increase is that the evolution cycle will be slower. It means that in a fix space of time the larger is the population size the less is the number of generations evolved. This parameter has direct correlation with the parameter MNG since it affects the total generations on a given time period. Varying this parameter in the range between 10 and 150 in steps of 20 showed that the best results are found with the value 50. This parameter suits well to the whole set of problems and its variation on a narrow range in the pursuit of the best tuning presented no benefits.

4.5 Initial Density – IND

During the initial population generation the probability used to attribute a "1" value to each gene is exactly the parameter IND. It influences the density of all individuals of the first population generation. However the density of each individual is not guaranteed to have the exact value of IND. The mean population density is expected to be probabilistically approximated to IND. This is not a relevant issue since small variations on the individual densities does not affect this initial phase significantly. The value itself is actually very important. If IND is too low the first generation evaluation is faster but the mean population fitness is low. On the opposite direction high values of IND tends to dispend more time on the evaluation phase of the first generation but leads to better mean population fitness. The tuning procedure consists in finding the ideal value of IND that balances efficiency and efficacy to each problem instance since they have a wide range of complexity. Each problem instance demands a particular value accordingly to its preliminary test execution times and fitness results. This parameter is closely related to IDD described right bellow.

4.6 Solver Timeout – ST

We had a problem that happens eventually but used to affect adversely. Some individuals with high density could waste a much longer time to be solved (and thus evaluated) than the others. The cost-benefit of these individuals is prohibitive. To avoid this time waste a solver timeout mechanism was implemented. The solver has an amount of ST seconds to finish its execution. If the execution time exceeds ST a timeout occurs and the solver execution is aborted. In this situation the individual has a '0' fitness value attributed because its computation is considered unfeasible in the tolerable time interval. Due to the implementation of the chosen solver product the precision of this control is low for small time intervals. The solver opens time windows to our implemented timeout evaluation based on its internal execution logic. This time windows are non deterministic and they vary widely. This behavior is not critical. The executions subject to intervention are exaggerated outliners. Its evolution times have often an order of magnitude above the normal ones. A mean delay of about 20 seconds is not significant compared to tens of minutes of these outliners.

5. The Mathematical Model

To formulate the Constrained Two-dimensional Non-guillotine Cutting Problem, we resorted an Integer Linear Programming model proposed in Nepomuceno et. al. (2007). Consider a set of items grouped into m types. For each box type i, characterized by its length and width (li, wi), and value vi, there is an associated number of boxes bi. Consider also a large object, in this instance, a pallet, that has (L, W) as its length and width dimensions respectively. The items should be cut orthogonally from the pallet. Each 0-1 variable u_{ide} alludes to the decision of whether to cut or not an item of type i at the coordinate (d, e). u_{fdef} = 1, if an item with type I is allocated on the position (d, e); 0 otherwise. The constants d and e belongs respectively to the followings discretization sets:

$$X = \{x \mid x = \sum_{i=1}^{n} \alpha_i l_i, x \leq L - \min\{l_i, 1 \leq i \leq m\}, \alpha_i \in \mathbb{Z}^+\}; \quad (5)$$

$$Y = \{y \mid y = \sum_{i=1}^{m} \beta_i w_i, y \leq W - \min\{w_i, 1 \leq i \leq m\}, \beta_i \in \mathbb{Z}^+\}. \quad (6)$$

To avoid box overlap, the incidence matrix g_{idepq} is defined as:

$$g_{idepq} = \begin{cases} 1, \text{if } d \leq p \leq d + l_i - 1 \text{ and } e \leq q \leq e + w_i - 1 \\ 0, \text{otherwise.} \end{cases} \quad (7)$$

The Constrained Two-dimensional Non-guillotine Cutting Problem can be formulated as:

$$\max \quad \sum_{i=1}^{m} \sum_{d \in X} \sum_{e \in Y} v_i u_{ide} \quad (8)$$

Subject to

$$\sum_{i=1}^{m} \sum_{d \in X} \sum_{e \in Y} g_{idepq} u_{ide} \leq 1, \forall p \in X, \forall q \in Y \quad (9)$$

$$\sum_{d \in X} \sum_{e \in Y} u_{ide} \leq b_i, 1 \leq i \leq m \quad (10)$$

$$u_{ide} \in \{0, 1\}, 1 \leq i \leq m, \forall d \in X, \forall e \in Y$$

(11)

6. Computational Results

For computational tests, we used the data obtained from Farago and Morabito (2000) for table 1 and table 2. It was used just one type of box for each problem instance. The data has to kind of pallet problem. One kind are problems from the literature and the other one are problems obtained from a carrier company. It was executed ten instances of the literature problem and thirty instance of a carrier company. The model was implemented in Java language, using the library Concert22 to link it to the solver. The solver used was CPLEX10.0 developed by ILOG Company. The tests ran in an Intel Core 2 Quad, with 8 Gb of memory.

The table 1 shows the results of the literature problem. This table is composed by the number of the problem instance, the pallet's dimension (W, L) followed by the box's dimension (w, l), the optimal solution for the given instance, the solution found, and the time elapsed to solve the problem.

Instance	Pallet (L,W)	Box (l, w)	Optimal Solution	Obtained Solution	Time elapsed (seg)
L1	22x16	5x3	23	21	5,755
L2	86x82	15x11	42	39	14,443
L3	43x26	7x3	53	48	11,108
L4	87x47	7x6	97	84	31,598
L5	42x39	9x4	45	40	17,664
L6	124x81	21x10	47	43	11,548
L7	40x25	7x3	47	46	22,698
L8	52x33	9x4	47	45	17,373
L9	57x44	12x5	41	37	10,673
L10	56x52	12x5	48	40	17,166
Average	-	-	49	44,3	16,0026

Table 1. Computational results obtained from literature problems.

The table 2 shows the results obtained from the Carrier Company instances. It is shown the number of the problem instance, the box's dimension (w, l), the optimal solution for the instance, the obtained solution using our approach, and the time elapsed while solving the problem. In this case, the pallet's dimension is fixed.

Instance	Box (l,w)	Optimal Solution	Obtained Solution	Time elapsed(seg)
R1	31x22	16	16	3,528
R2	50x20	12	12	2,517
R3	33x23	15	15	1,436
R4	34x26	12	11	4,649
R5	36x15	20	20	5,641
R6	28x21	19	17	5,661
R7	32x18	20	18	5,536
R8	38x26	10	10	1,096
R9	25x15	32	32	2,805
R10	46x30	8	8	0,359
R11	39x25	12	12	0,514
R12	38x20	15	15	1,031
R13	49x20	12	12	0,716
R14	28x17	23	23	4,279
R15	40x29	10	9	3,12
R16	35x12	26	26	1,749
R17	27x22	19	17	7,501
R18	21x12	#	45	19,621
R19	24x19	26	25	5,972
R20	32x24	15	15	0,886
R21	26x20	22	22	1,498
R22	19x14	43	41	13,294
R23	44x29	8	8	0,731
R24	52x33	6	6	0,175
R25	36x21	15	14	4,954
R26	35x20	15	15	0,334
R27	20x14	42	40	12,833
R28	22x17	31	28	10,984
R29	37x20	15	15	1,33
R30	24x13	38	36	11,801
Average	-	19,20689655	19,43333333	4,5517

Table 2. Computational results obtained from Carrier Company problems.

7. Conclusions

As it can be seen, the use of Hybrid Methodology showed significant results, near from optimal in a reasonable time. In some instances, there still some area of the total that should fit another box, but it does not happen due to the shape of the boxes on its width and/or length that does not actually fits on the pallet. As it was said before, the pallet is used in a large scale for storage, and it is produced over five million pallets a year. It is known that there is a cost for storage, so, the best the usage of the storage area, the less is the cost to maintain the items.

The box's dimension is relevant to the problem. It is reasonable to have default patterns of boxes to fill the pallet on the best manner as possible. Accordingly to the obtained results, the more irregular is the shape box's shape, the less is the used area of the pallet. But it can have some bigger shapes that could fit better than the shape of the boxes used. There are many approaches to reduce storage cost, and the Hybrid Methodology proved useful for this kind of problem.

For future works, it is intended to use new approaches in order to obtain better results and get even closer from the optimal solution.

References

Beasley J. E.. A Population Heuristic for Constrained Two-dimensional Non-guillotine Cutting, volume 156, chapter European Journal of Operational Research.2004.

Chu P. C., A Genetic Algorithm Approach for Combinatorial Optimization Problems. PhD thesis, 1997.

Cook W. and Seymour P.. Tour Merging via Branch-Decomposition, volume 15, chapter INFORMS Journal on Computing, 2003.

Dowsland K. A. and Dowsland W. B., Packing Problems, volume 56, chapter European Journal of Operational Research, pages 2-14. 1992.

Farago R. and Morabito R., Um Método Heurístico Baseado em Relaxação Lagrangiana para o Problema de Carregamento de Paletes do Produtor, Pesquisa Operacional, Vol.20 no.2 Rio de Janeiro Dec. 2000.

Fekete S. P. and Schepers J., A New Exact Algorithm for General Orthogonal D-dimensional Knapsack Problems, volume 1284, chapter Proceedings of the 5th Annual European Symposium on Algorithms, LNCS, pages 144 - 156. Springer - Verlag, 1997.

Gehring H., Menschner K., and Meyer M., A computer-based heuristic for packing pooled shipment containers, volume 44, chapter European Journal of Operational Research, pages 277 - 288, 1990.

Hadjiconstantinou E. and Christodes N., An Exact Algorithm for General, Orthogonal, Two dimensional Knapsack Problems, volume 83, chapter European Journal of Operational Research, pages 39 - 56. 1995.

Kocjan W. e Holmström K., Generating Stable Loading Patterns for Pallet Loading Problems, Applied Mathematics, CPAIOR 2008: The Fifth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 2008.

Mahfoud S. W. and Goldberg D. E., Parallel Recombinative Simulated Annealing: A Genetic Algorithm, volume 21, chapter Parallel Computing, 1995.

Martin O., Otto S. W., and Felten E. W., Largestep Markov Chains for the TSP: Incorporating Local Search Heuristics, volume 11, chapter Operations Research Letters, 1992.

Nepomuceno N., Pinheiro P. R., and Coelho A. L. V., Tackling the container loading problem: A hybrid approach based on integer linear programming and genetic algorithms. In EvoCOP 2007 - Valencia, Spain. LNCS, volume 4446, pages 154 -165. Springer-Verlag, 2007a.

Nepomuceno N., Pinheiro P. R., and Coelho A. L. V., Combining Metaheuristics and Integer Linear Programming: A Hybrid Methodology Applied to the Container Loading Problem. In: XXVII Congresso da Sociedade Brasileira de Computação, 2007, Rio de Janeiro. Anais do XXVII Congresso da Sociedade Brasileira de Computação, p. 2028-2032, 2007b.

Nepomuceno N., Pinheiro P. R., and Coelho A. L. V., A hybrid optimization framework for cutting and packing problems: Case study on constrained 2D non-guillotine cutting, vol. Recent Advances in Evolutionary Computation for Combinatorial Optimization of Studies in Computational Intelligence, chap. 6, Springer-Verlag, 87–99, 2008.
Nepomuceno N., Pinheiro P. R., and Coelho A. L. V. Aplicação de uma Metodologia Híbrida ao Problema de Carregamento de Contêiner. In: XXXVIII Simpósio Brasileiro de Pesquisa Operacional, 2006, Goiania. Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional. Rio de Janeiro, 2006. p. 1596-1603
Puchinger J. and Raidl G., Combining Metaheuristics and Exact Algorithms in combinatorial Optimization: A Survey and Classication, volume 3562, chapter Proceedings of the 1st International Work-Conference on the Interplay Between Natural and Articial Computation, LNCS, pages 41-53. Springer-Verlag, 2005.
Raidl, G. R., A Unified View on Hybrid Metaheuristics, vol. 4030. Springer-Verlag, 2006, ch. Proceedings of the 3rd International Hybrid Metaheuristics Workshop, LNCS, pp. 1-12.
Talbi E. G., A Taxonomy of Hybrid Metaheuristics, volume 8, chapter Journal of Heuristics, pages 541-564. 2002.
Vasquez M. and J. Hao K.. A Hybrid Approach for the 0-1 Multidimensional Knapsack Problem, chapter Proceedings of the 17th International Joint Conference on Articial Intelligence, 2001.