

CYLINDER PACKING BY SIMULATED ANNEALING

M. Helena Correia

INESC PORTO – Instituto Engenharia de Sistemas e
Computadores do Porto – Portugal
email: mcorreia@inescporto.pt

José F. Oliveira

J. Soeiro Ferreira

FEUP – Faculdade de Engenharia da Universidade
do Porto
INESC PORTO – Instituto Engenharia de Sistemas e
Computadores do Porto – Portugal

Abstract

This paper is motivated by the problem of loading identical items of circular base (tubes, rolls, ...) into a rectangular base (the pallet). For practical reasons, all the loaded items are considered to have the same height. The resolution of this problem consists in determining the positioning pattern of the circular bases of the items on the rectangular pallet, while maximizing the number of items. This pattern will be repeated for each layer stacked on the pallet. Two algorithms based on the meta-heuristic Simulated Annealing have been developed and implemented. The tuning of these algorithms parameters implied running intensive tests in order to improve its efficiency. The algorithms developed were easily extended to the case of non-identical circles.

Keywords: cylinder packing, combinatorial optimization, simulated annealing.

Resumo

Este artigo aborda o problema de posicionamento de objetos de base circular (tubos, rolos, ...) sobre uma base retangular de maiores dimensões. Por razões práticas, considera-se que todos os objetos a carregar apresentam a mesma altura. A resolução do problema consiste na determinação do padrão de posicionamento das bases circulares dos referidos objetos sobre a base de forma retangular, tendo como objetivo a maximização do número de objetos estritamente posicionados no interior dessa base. Este padrão de posicionamento será repetido em cada uma das camadas a carregar sobre a base retangular. Apresentam-se dois algoritmos para a resolução do problema. Estes algoritmos baseiam-se numa meta-heurística, *Simulated Annealing*, cuja afinação de parâmetros requereu a execução de testes intensivos com o objetivo de atingir um elevado grau de eficiência no seu desempenho. As características dos algoritmos implementados permitiram que a sua extensão à consideração de círculos com raios diferentes fosse facilmente conseguida.

Palavras-chave: empacotamento de cilindros, otimização combinatória, *simulated annealing*.

1. Introduction

Stocking, manipulation and distribution procedures of “small” items are important phases of several industrial activities. In this context, the problem of positioning small items inside bigger spaces, so that unused space is minimized, often arises. It is generally known that the effectiveness of these procedures can be a quite important economical factor for the success of these industries.

The positioning problem tackled in this paper is usually referred to as the “pallet loading problem”. This is intrinsically a three dimensional packing problem which most frequently consists in finding efficient positioning patterns of identical objects (usually, rectangular or cylindrical objects) on a rectangular base (pallet), where the vertical orientation of the objects is determined by practical constraints. These patterns are repeated for each layer stacked on the pallet. Therefore, the standardization of the objects vertical dimension supports the reduction of the dimensionality of the problem, and so two dimensional techniques may be applied to maximize the number of identical pieces (objects base) inside a rectangle (pallet) without overlapping.

The “cylinder packing problem” or “cylinder palletization” is basically concerned with the densest packing of identical circles (cylinders base) inside a rectangle (the pallet). It can be considered a NP-hard problem since it is a generalization of the knapsack problem [Gar79] and, so, it is very unlikely that a polynomial time algorithm can be developed to solve it.

There is not much published literature on this subject and, due to its complexity, the approaches to solve the problem are mainly heuristics: in [Ise91], Isermann proposes some heuristic techniques for packing identical circles in homogeneous patterns developed geometrically, restricting its attention to problems where the product is small relative to the rectangle; in [Dow91], Dowland addresses the related problem of deciding the optimal package size in which to stow identical cylindrical objects with fixed orientation; and in [Fra94], Fraser and George discuss loading reels inside a container in the context of a paper industry, where the circles relative position is chosen among a set of easily stowed pre-defined patterns. Most recently published works also consider the problem of packing circles of different sizes inside a rectangle: in [Hif94], Hifi, Pachos and Zissimopoulos develop a Simulated Annealing based approach to solve the generic “circular cutting problem”, where the density of the circles placed in an open-strip is considered to solve the problem; and in [Geo95], George, George and Lamar discuss several heuristic approaches and include stability considerations in the context of loading pipes of different diameters inside a container. This latter also proposes a mixed integer non-linear formulation for the problem of fitting different circles inside a rectangle which, according to our experience, is difficult to solve, even for a small number of pieces, by the current general purpose mixed integer non-linear optimization software packages. More recently, Stoyan and Yaskov [Sto98] propose a solution method based on a branch-and-bound algorithm and a reduced gradient method to solve a mathematical model of the problem of placement of rectangles and circles in a larger rectangle. A more analytical line of work, considers the related problem of maximizing the diameter of equal non-overlapping circles contained in a unit square (which is equivalent to maximize the minimum pairwise distance among n points in a unit square). An overview of the bibliography, from 1965 till 1995, addressing this more restrictive problem can be found in [Mar95].

This paper proposes a Simulated Annealing approach to the problem of packing identical circles inside a rectangle. Simulated Annealing is a general-purpose local search algorithm

based on the analogy between the physical annealing of solids and combinatorial optimization problems. In spite of the good results achieved in a wide variety of contexts, Simulated Annealing is “in no way a panacea”. Its performance depends not only on its adequacy for the problem to solve but also on the formulation adopted and on the low level algorithms chosen to generate solutions and evaluate the objective function. High level heuristics such as Simulated Annealing (usually called meta-heuristics) are strongly dependent on implementation decisions concerning problem specific choices (objective function, solution space, neighborhood structure and initial solution) and generic choices (initial temperature value, cooling schedule, acceptance probability and stopping criterion). More details about this technique can be found in [Pir92], [Ree95] and [Vid93].

The paper is organized as follows. Section 2 introduces two Simulated Annealing based algorithms to solve the problem of fitting circles inside a rectangle. These algorithms basically differ in both the neighborhood structure and the objective function adopted. The MinOSA algorithm, described in section 2.1, considers the arrangement which minimizes the overlap among a fixed number of circles inside a rectangle. The MaxNSA algorithm in section 2.2 allows variation in the number of circles during the heuristic process and tries to establish a trade-off between the maximization of the number of circles and the minimization of the overlap among those circles in the final solution. Each of the developed implementations of Simulated Annealing required an additional experimental effort to tune up the parameters involved.

The two algorithms, MinOSA and MaxNSA, are the core of two other higher level algorithms, CPA-MinOSA and CPA-MaxNSA, respectively. These two algorithms are intended to solve the cylinder packing problem which solutions should maximize the number of circles packed without overlapping (see sections 2.3 and 2.4). The computational results obtained with these algorithms for a varied set of problem instances are discussed in section 3. This section also compares the Simulated Annealing based implementations with two other pure local search algorithms either in terms of speed (efficiency) and in terms of the solution quality (efficacy). Finally, CPA-MaxNSA is extended for packing non-identical circles.

2. Cylinder Packing Algorithms

The algorithms implemented for the resolution of the cylinder packing problem can be divided in two layers. The lower layer consists in applying Simulated Annealing to determine the position of identical circles inside a rectangle while minimizing overlap among them. In this scope, two algorithms were implemented, MinOSA and MaxNSA, which basically differ in the neighborhood structure and objective function adopted. MinOSA considers a pre-defined number of circles randomly positioned inside the rectangle, and tries to minimize overlap among these circles. In MaxNSA the number of circles is not fixed and the algorithm tries both to maximize the number of circles in the final solution and minimize overlap among these circles.

The previous algorithms do not guarantee solutions without overlap among the circles, which is a basic constraint of the cylinder packing problem. They are both used by two other algorithms, CPA-MinOSA and CPA-MaxNSA, to solve this positioning problem. The way this higher-level algorithms use the two Simulated Annealing based algorithms, MinOSA and MaxNSA, will be described in this section and is schematized in Fig. 1.

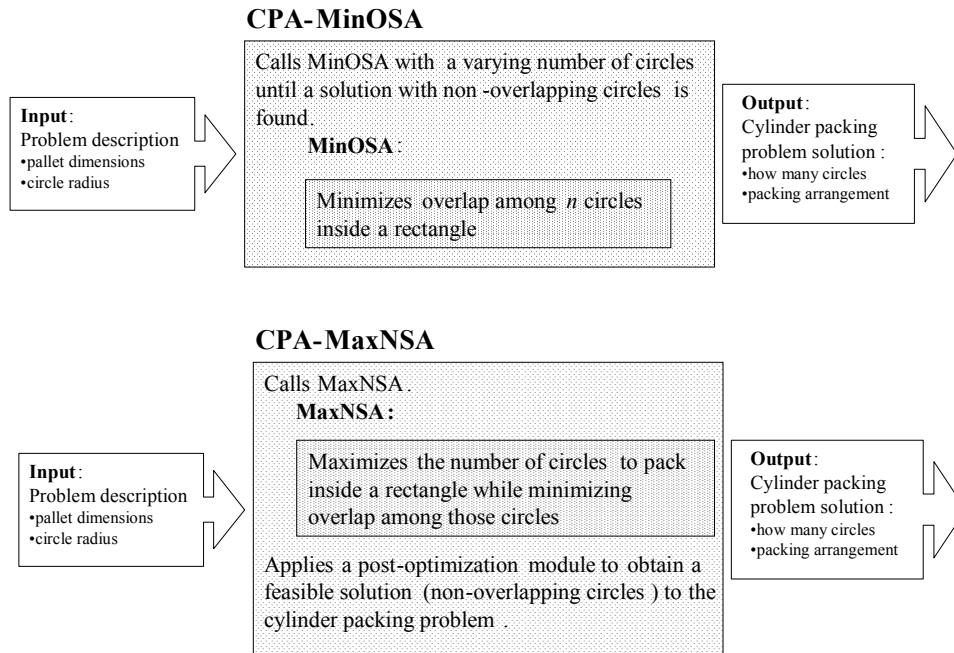


Figure 1 – CPA-MinOSA and CPA-MaxNSA

2.1 Minimizing Overlap with Simulated Annealing (MinOSA)

The first Simulated Annealing algorithm implemented was intended to determine the layout of a pre-determined number of identical circles inside a rectangle while minimizing the overlap among those circles (MinOSA). This implementation required the definition of a discrete set of solutions and so, a position grid was considered on the pallet restricting both the coordinates of the circles inside the rectangle and the dimensions of the objects to integer values. This algorithm starts with a solution which consists of a pre-defined number of circles randomly placed inside the rectangle and without crossing the pallet borders. A neighbor is obtained by moving a randomly chosen circle of one grid unit in one of the eight possible directions marked in Fig. 2. Once again, this movement is restricted by the pallet borders. In this neighborhood structure, the neighbor solutions are obtained from the current solution by applying to each circle one of the eight movements described above. A uniform probability distribution determines the selection of the circle to move and the movement to apply.

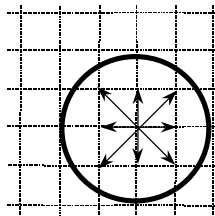


Figure 2 – Neighborhood Structure (MinOSA)

The objective function ($F(x)$) adopted in MinOSA evaluates the amount of overlap among the circles in a solution (x). However, due to the calculation complexity of this function, an approximated objective function, with a similar behavior to the optimal one, was adopted. This approximation considers the square hull of the circles instead of the circles themselves, being computed the intersection of the square hulls when two circles overlap (the distance between the two centers is smaller than the sum of the circles radii). This function is easier to evaluate and results in an improvement of the algorithm efficiency.

The generic decisions assumed in the implementation of MinOSA were based on the results obtained from a set of experimental tests and were intended to establish a trade-off between the algorithms efficacy and efficiency. These tests were based on three instances in which the circle area is approximately 5%, 10% and 15% of the pallet area, respectively. Due to the random nature of the algorithm, the averages of the results obtained in 10 computational runs of each test instance were considered. The number of circles used in each test instance was determined by the upper bound introduced in [Cor00]. A constructive methodology was adopted for the parameters tuning process in which each parameter was tuned using the good values already identified for previously tuned parameters. In fact, this is a simplified process as it ignores the correlation between the several parameters involved, but the improvement on the algorithm performance was quite satisfactory. The decisions concerning the parameters are the following, in the order they were considered:

- The **acceptance probability function** adopted is based on the Boltzmann distribution:

$$p(\Delta F_n, T_n) = e^{-\frac{\Delta F_n}{T_n}}$$

where $\Delta F_n = F(x_n) - F(x_{n-1})$ and T_n corresponds to the temperature at iteration n .

- The geometric **cooling schedule** was adopted:

$$T_n = \alpha T_{n-1}$$

Parameter α was tested considering values between 0.90 and 0.99. $\alpha=0.97$ was found the most adequate. The initial temperature was determined based on the probability of acceptance of the worst solution (p_0). Values of 0.01, 0.05, 0.1 and 0.5 for p_0 were tested and $p_0=0.1$ was finally adopted.

- The **number of iterations at each temperature** (L_n) was assumed fixed and was based on the average size of the neighborhood:

$$L = n \times \text{average_size_of_the_neighborhood_structure}$$

Values of n between 1 and 10 were tested and finally n was set to 4.

- The **stopping criterion** adopted was based on the number of consecutive temperatures without improvement on the objective function value. Initially this value was set to 10, but the need to improve the algorithm efficiency determined 6 as the number of temperatures which leads to satisfactory solutions without excessive time consumption.

2.2 Maximizing the Number of circles with Simulated Annealing (MaxNSA)

The second algorithm implemented differs from the previous one basically by the specific decisions concerning the neighborhood structure and the objective function.

This Simulated Annealing based algorithm, MaxNSA, deals with a variable number of circles during its execution and tries to establish a trade-off between the maximization of the number of circles packed inside the rectangle and the minimization of the overlap among those circles.

As previously, the initial solution consists in several circles randomly placed inside the rectangle without crossing its borders. The upper bound introduced in [Cor00] is used to determine the initial number of circles. This number is also used to limit the number of circles simultaneously packed.

Besides the movements considered by the neighborhood structure of MinOSA, this algorithm also considers the possibility of removing a randomly chosen circle and of including a new one randomly placed inside the rectangle (see Fig. 3). This neighborhood structure leads to the reformulation of the objective function in order to include the maximization of the number of circles packed:

$$\min F(x) = \text{overlap}(x) - \beta \times \text{total_circles_area}(x) \quad [1]$$

Coefficient β weights the second component of the objective function in order to distinguish those situations in which the inclusion of one circle causes identical variations in the two components of the objective function. The value of β must be in the interval $]0,1[$. Its concrete value must be tuned and should give privilege to solutions with less overlap, once this is a requirement to solve the cylinder packing problem.

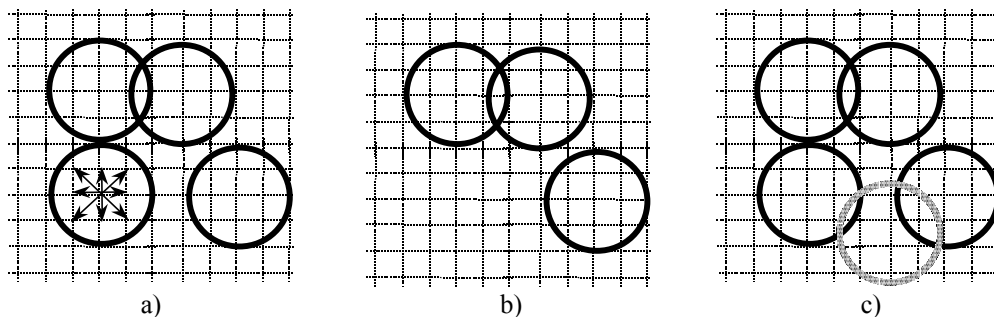


Figure 3 – Neighborhood Structure (MaxNSA).
a) Movement of a circle; b) Removal of a circle; c) Inclusion of a circle

The need to tune the algorithm parameters also arose in the scope of this implementation. The methodology adopted here was identical to the one described previously for MinOSA and, in most cases, the final parameters values came out to be quite similar to those found in the previous algorithm. The main differences concern the following parameters:

- Due to the variable size of the neighborhood structure in this algorithm, the definition of the **number of iterations at each temperature** (L_n) is based on the acceptance rate of worse solutions at each temperature. When the pre-defined threshold value for this acceptance rate is attained the temperature is lowered. Values between 0.1 and 0.5 were tested and finally the acceptance rate of worse solutions was experimentally set to 0.4.
- The **stopping criterion** adopted was also based on the number of consecutive temperatures without improvement on the objective function value. Initially this value was set to 10, but the need to improve the algorithm efficiency lowered this value to 5.

- Finally, parameter β had also to be tuned. In the previous tests, this parameter was set to 0.5. Then it was tested with values between 0.1 and 0.9. The results seem to support a correlation between the value adopted for β and the relative dimensions of the circle and the pallet. However, no reasoning was found for this correlation and a fixed $\beta=0.5$ was adopted. In fact, the initial value used to tune the other parameters was also considered to be the most adequate in average.

As before, the algorithm efficiency was clearly improved by parameter tuning.

2.3 Cylinder Packing Algorithm using MinOSA (CPA-MinOSA)

The first cylinder packing algorithm that will be described consists in successively applying the Simulated Annealing based algorithm MinOSA to instances differing by one circle in each successive execution of the algorithm. Two strategies can be adopted:

- the initial number of circles is guaranteed to be larger than or equal to the number of circles in the optimal solution (i.e., an upper bound for the number of circles) and it is decreased of one unit in successive executions of MinOSA until a solution with no overlap among the circles is attained;
- the initial number of circles is guaranteed to be smaller than or equal to the number of circles in the optimal solution (i.e., a lower bound for the number of circles) and it is increased of one unit in successive executions of MinOSA until it is not possible to obtain a solution without overlapping circles; the solution which packs the maximum number of circles without overlap is considered the final solution.

The CPA-MinOSA algorithm implemented is shown in Fig. 4 and follows the second approach mentioned above, as it seems to be more efficient. In fact, the first approach is much dependent on the quality of the upper bound considered and, if it is not close to the optimal value, it can be quite expensive in terms of computational time. On the contrary, a good lower bound to the optimal number of circles is not so crucial as the MinOSA stops as soon as a non-overlapping solution is found.

```

algorithm CPA-MinOSA;
  number_circles = lower_bound();
  repeat
    sol = MinOSA(number_circles);
    if overlap(sol) == 0 then
      final_sol = sol;
      number_circles = number_circles + 1;
  while overlap(sol) == 0;

```

Figure 4 – Algorithm CPA-MinOSA

2.4 Cylinder Packing Algorithm using MaxNSA (CPA-MaxNSA)

The cylinder packing algorithm CPA-MaxNSA is, in the scope of this work, coincident with MaxNSA. Although MaxNSA does not guarantee a feasible solution to cylinder packing as it does not assure the in-existence of overlap among the circles packed, its good performance in terms of computational time allows the inclusion of a post-optimization module in order to guarantee solutions feasibility. The performance of this post-optimization heuristic will be

determinant in the quality of the final solutions to the cylinder packing problem, but its discussion was considered to be out of the scope of this paper.

3. Computational Results

The computational tests described in this section evaluate the algorithms performances and establish comparative analyses between them and with two pure local search algorithms [Vid93]: LS1- a local search algorithm with selection of the best neighbor, and LS2- a local search algorithm with selection of the first better neighbor. This was intended to clarify the advantages introduced by a less greedy and more exhaustive approach such as Simulated Annealing.

The data used in these computational experiments is described in Table 1. In order to identify typical behaviors concerning the relative size of the objects involved, the test instances were generated such as to be representative of three possible relations between the pallet area and the circle area: in data sets 1.x, 2.x and 3.x the circle area is approximately 5%, 10% and 15% of the pallet area, respectively. We also tried to diversify the test instances in terms of the pallet shape and so, in data sets x.1, x.2 and x.3, the pallet shapes considered are rectangular-long, rectangular-short and square, respectively. This table also includes an upper bound and lower bound on the number of circles, introduced in [Cor00].

Table 1 – Test Instances

Data	Pallet Dim.	Circle Radius	Upper Bound	Lower Bound
1.1	100×200	18	17	15
1.2	120×80	12	18	15
1.3	100×100	13	16	12
2.1	100×200	25	8	8
2.2	120×80	17	9	6
2.3	100×100	18	8	6
3.1	100×200	31	5	3
3.2	120×80	21	6	4
3.3	100×100	22	5	4

C programming language was used to implement the algorithms and the computational tests were performed on a Pentium MMX-200MHz. The algorithms evaluation focused on features such as objective function value and overlap of the final solution, running time and number of circles packed. Due to the random nature of Simulated Annealing, the results in this section are all based on average values of 10 runs for each test instance. These experiments were intended to:

- evaluate MinOSA and MaxNSA (CPA-MaxNSA) performances and compare them with two pure local search algorithms;
- compare the two cylinder packing algorithms, CPA-MinOSA and CPA-MaxNSA;
- compare the results obtained with other published results.

Finally, an extension of CPA-MaxNSA to the packing of non-identical cylinders is introduced.

3.1 MinOSA

The MinOSA algorithm was tested with a larger set of instances considering the best parameters values found in section 2.1. The MinOSA performance in terms of the objective function value (overlap) and computing time is pictured in Fig. 5. The number of circles considered in each instance is the upper bound for the number of circles (see Table 1) and so, it was not always possible to reach solutions without overlap.

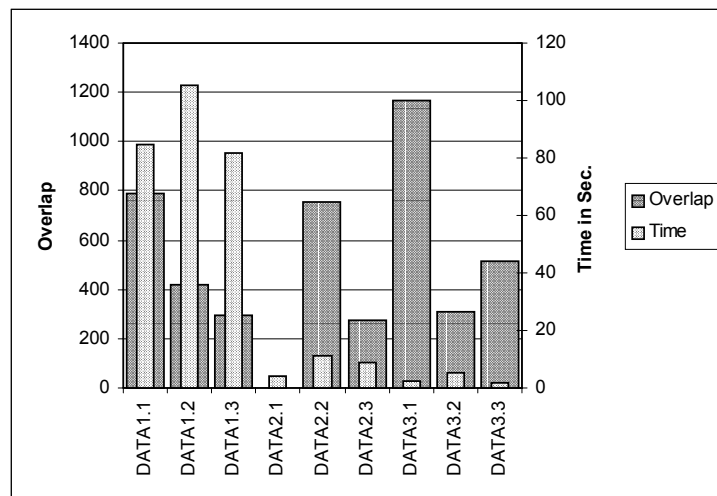


Figure 5 – MinOSA performance

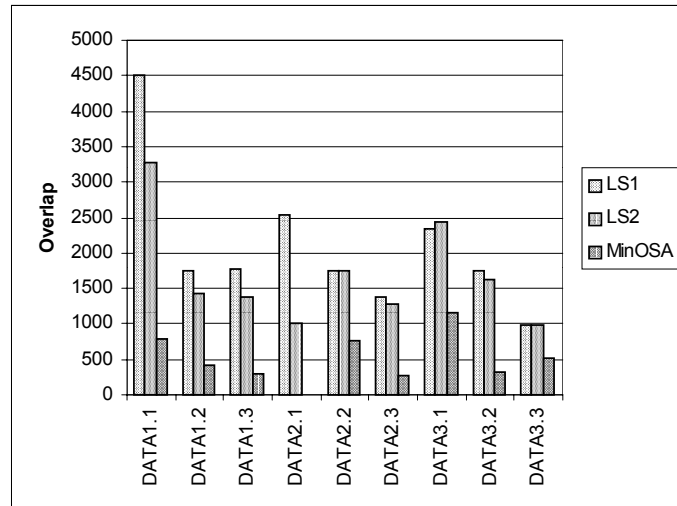
From the results stands out the greater efficiency of the algorithm in terms of computational time when the ratio *pallet area/circle area* is smaller (i.e., when less circles are involved). This is due to a smaller neighborhood which directly determines the number of iterations to perform at each temperature. Table 2 shows the average time taken to solve 1000 iterations for each instance considered.

Table 2 – Average time taken by MinOSA to perform 1000 iterations

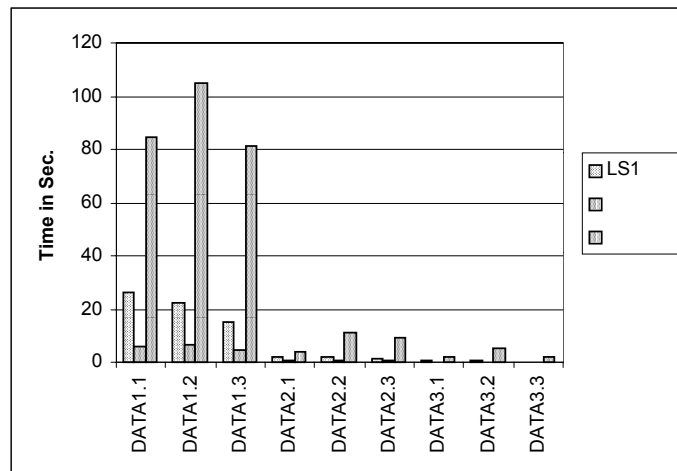
Data	Av. time per 1000 iterations (in sec.)
1.1	1.187
1.2	1.301
1.3	0.969
2.1	0.590
2.2	0.396
2.3	0.295
3.1	0.187
3.2	0.225
3.3	0.217

In Fig. 6.a) and b), MinOSA is compared with the other two local search algorithms, LS1 and LS2, in terms of the overlap in the final solution and of the running time required to

reach it. The improvement introduced by the Simulated Annealing based algorithm in terms of objective function is achieved with a considerable increase of execution time, if we consider those instances with larger ratio *pallet area/circle area*.



a)



b)

Figure 6 – MinOSA vs Local Search Algorithms. a) Overlap ($F(x)$); b) Execution time

3.2 MaxNSA

In Fig. 7, MaxNSA is evaluated in terms of computing time and overlap in the final solution (which differs from the objective function value used in the search – see expression 1). As in the previous section, the algorithm parameters were set accordingly to the best values found

during the tuning process. The results reveal the greater efficacy of MaxNSA when the circles are smaller opposing to a higher consumption of execution time by those instances. The average time taken by this algorithm to solve 1000 iterations for each test instance is presented in Table 3.

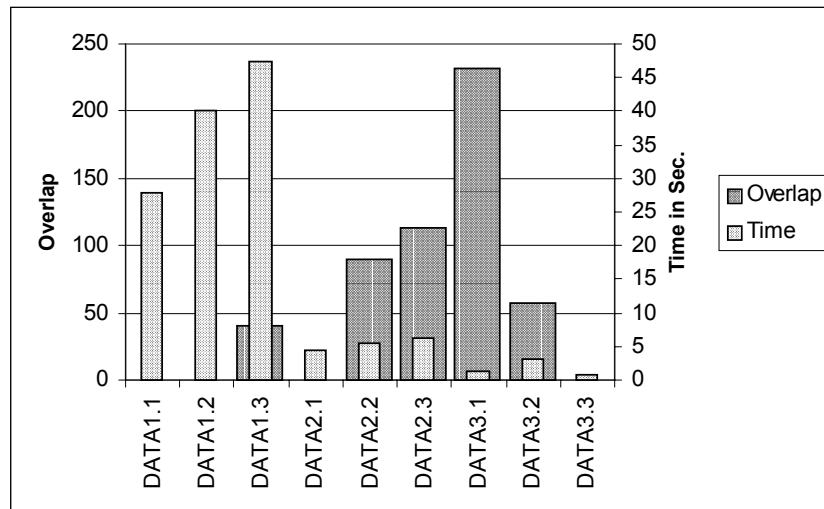


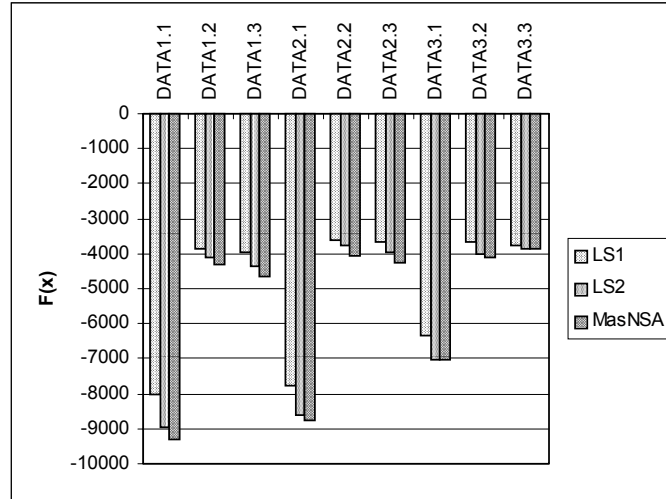
Figure 7 –MaxNSA performance

Table 3 – Average time taken by MaxNSA to perform 1000 iterations

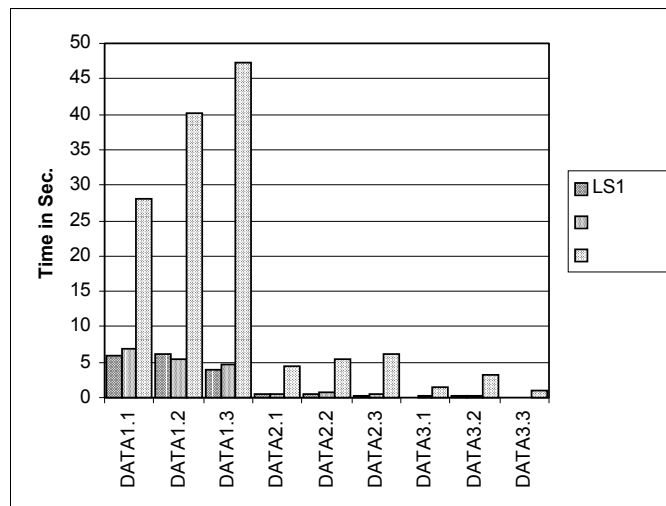
Data	Av. time per 1000 iterations (in sec.)
1.1	0.665
1.2	0.746
1.3	0.677
2.1	0.209
2.2	0.256
2.3	0.283
3.1	0.105
3.2	0.138
3.3	0.114

The significant amount of overlap in those instances with smaller number of circles must not be considered as an indicator of the inadequacy of the algorithm to the resolution of the cylinder packing problem. They merely reflect the algorithm sensitivity to the value assumed by coefficient β in the objective function (see expression 1).

In Fig. 8.a) and b), MaxNSA is compared with the two local search algorithms LS1 and LS2. From the results stands out the smaller relative improvement introduced by MaxNSA, if we consider the comparative results for MinOSA. Although, this small variation in the objective function value may be determinant in the solution feasibility to the cylinder packing problem.



a)



b)

Figure 8 – MaxNSA vs Local Search Algorithms. a) Overlap ($F(x)$); b) Execution time

3.3 CPA-MinOSA vs. CPA-MaxNSA

In this section, the two cylinder packing algorithms, CPA-MinOSA and CPA-MaxNSA are compared. As stated in section 2.4, the algorithm CPA-MaxNSA may not provide a feasible solution to the cylinder packing problem as the in-existence of overlap in the final solution is not guaranteed.

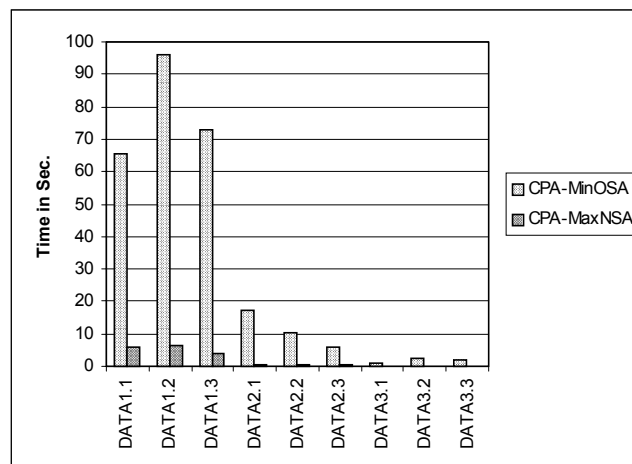
The algorithms were compared in terms of the maximum number of circles packed without overlapping and the occurrence rate of the best solution in the 10 runs of each test instance (Table 4).

Table 4 – CPA-MinOSA vs CPA-MaxNSA

Data	CPA-MinOSA		CPA-MaxNSA	
	N. of Circles Packed	Occur. Rate	N. of Circles Packed	Occur. Rate
1.1	15	1	15	1
1.2	15	1	15	1
1.3	13	1	0	-
2.1	8	1	7	1
2.2	7	1	7	0.1
2.3	6	1	0	-
3.1	3	1	0	-
3.2	4	1	4	0.1
3.3	4	1	4	1

These results reveal the robustness of CPA-MinOSA, since the rate of occurrence of the best solution is 1 for every test instance considered. The greater efficacy of this algorithm stands on the fact that it always packs at least as much non-overlapping circles as CPA-MaxNSA.

The comparative evaluation of the efficiency of the two cylinder packing algorithms, CPA-MinOSA and CPA-MaxNSA, was based on their execution time to solve the test instances. The results obtained are shown in Fig. 9. As expected, CPA-MinOSA expends considerably more time to reach the final solution than CPA-MaxNSA. This is mainly due to the iterative process in which CPA-MinOSA is based (see Fig. 4).

**Figure 9** – CPA-MinOSA vs CPA-MaxNSA

Owing to the lack of effectiveness of CPA-MaxNSA, it may not be considered as a stand-alone method to solve the cylinder packing problem. In fact, the short time taken by this algorithm to reach the final solution seems to be its great attractive. This characteristic can be exploited by following the MaxNSA algorithm by a post-optimization module that eventually may re-position the circles and remove the exceeding ones.

The two cylinder packing algorithms were also compared with some results found in the literature. Two instances from [Ise91], *ISER1* and *ISER2* are used in this comparative analysis. The other instances considered, named *SQR_*, are from [Gol70], in which results

obtained for the packing of n ($n \leq 27$ as well as for some $n > 27$) identical circles into squares of minimum area are tabulated. Table 5 describes several instances considered and displays the results in which the algorithms evaluation was based. The *NumPubl* column includes the best published results for each instance and *NumCPA* shows the best result obtained with the cylinder packing algorithms implemented. Fig. 10 presents a layout obtained for some of the instances tested. The best solutions obtained with CPA-MinOSA were always at least as good as those obtained with CPA-MaxNSA. On the other hand, the algorithm CPA-MaxNSA failed to attain to any feasible solution for several test instances.

Table 5 – Test instances used in the comparative evaluation

Data	Rectangle Dimensions	Circle Radius	NumPubl	NumCPA
ISER1	1200×800	102	22	22
ISER2	1200×800	101	23	22
SQR2	100×100	29	2	2
SQR3	250×250	63	3	3
SQR4	100×100	25	4	4
SQR5	100×100	20	5	5
SQR6	100×100	18	6	6
SQR7	500×500	87	7	7
SQR8	200×200	34	8	8
SQR9	100×100	16	9	9
SQR10	300×300	44	10	10
SQR11	500×500	71	11	11
SQR12	800×800	112	12	11
SQR13	100×100	13	13	13
SQR14	500×500	64	14	14
SQR15	500×500	63	15	15
SQR16	500×500	62	16	16
SQR17	300×300	35	17	17
SQR18	600×600	69	18	17
SQR19	900×900	101	19	18

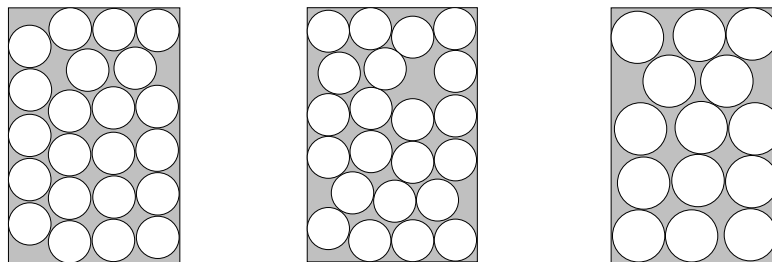


Figure 10 – Layout for three test instances

When comparing column *NumCPA* with column *NumPubl*, it can be observed that *NumCPA* equals the best results published in 85% of the test instances, being the solution different by one circle in the other 15% of the cases. Consequently, the main advantage of the Simulated Annealing based approach is its generality and simplicity of implementation, maintaining very good results.

Table 6 shows the results obtained for a larger number of instances, including instances of higher complexity than those previously considered. These results were obtained with CPA-MinOSA. The *UB* column lists an upper bound for each instance, computed as proposed by the authors in [Cor00].

Table 6 – Larger set of instances solved with CPA-MinOSA

Data	Rectangle Dimensions	Circle Radius	UB	CPA-MinOSA
1.1	160×80	6	101	90
1.2	100×200	8	88	84
1.3	120×240	10	81	73
1.4	100×80	5	90	86
1.5	120×80	6	75	68
1.6	120×100	6	94	87
1.7	80×80	5	72	68
1.8	100×100	6	78	70
1.9	120×120	7	83	73
2.1	160×80	10	35	32
2.2	100×200	13	33	29
2.3	120×240	15	35	32
2.4	100×80	8	34	32
2.5	120×80	9	33	28
2.6	120×100	10	33	30
2.7	80×80	7	37	32
2.8	100×100	9	34	30
2.9	120×120	11	33	29
3.1	160×80	14	18	15
3.2	100×200	18	17	15
3.3	120×240	21	18	15
3.4	100×80	11	18	16
3.5	120×80	12	18	15
3.6	120×100	14	17	14
3.7	80×80	10	17	16
3.8	100×100	13	16	13
3.9	120×120	15	17	16
4.1	160×80	20	8	8
4.2	100×200	25	8	8
4.3	120×240	30	8	8
4.4	100×80	16	8	6
4.5	120×80	17	9	7
4.6	120×100	20	8	6
4.7	80×80	14	9	6
4.8	100×100	18	8	6
4.9	120×120	21	9	6
5.1	160×80	25	5	3
5.2	100×200	31	5	3
5.3	120×240	37	5	3
5.4	100×80	19	6	4
5.5	120×80	21	6	4
5.6	120×100	24	5	4
5.7	80×80	17	6	4
5.8	100×100	22	5	4
5.9	120×120	26	5	4

It should be noticed that just a few of the solutions presented in column *CPA-MinOSA* are proved to be optimal, based on the comparison with the upper bounds. This gap reaches a maximum of 66.67% (in terms of the number of circles), being in average 21,95%. This is mainly due to the intrinsic complexity of the test instances considered.

3.4 CPA-MaxNSA with different sized circles

The algorithm CPA-MaxNSA was extended to consider non-identical circles. This approach to the cylinder packing problem has the advantage of allowing the inclusion of circles of different radius in a quite natural way: the different sized circles are randomly chosen with the purpose of maximizing the utilization of the pallet area. This extension is not so easy in the case of the CPA-MinOSA, as, in each execution of MinOSA, this algorithm would require a pre-defined number of each type of circles to proceed with the positioning process.

Some adjustments had to be made in the implementation of MaxNSA to include non-identical circles, mainly concerning the initialization of the algorithm – number and size of the starting circles and the parameters which depend on these features, such as the initial temperature. These values were defined arbitrarily: the starting circles are randomly chosen restricting their total area to the area of the pallet; the initial temperature is computed as before, based on the overlap of those circles. All the other algorithm parameters were kept from the single-sized version. The objective function considered here also consisted in both maximizing the area of the circles inside the rectangle and minimizing the overlap among those circles.

Fig. 11 shows the solution obtained for an instance based on an example from [Geo95].

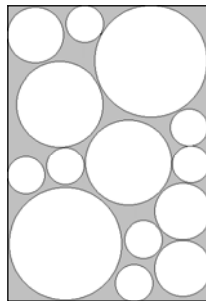


Figure 11 – Layout for three test instances

4. Conclusions and Future Development

This paper deals with the resolution of the problem of maximizing the number of identical cylinders packed in an upright position on a pallet – the cylinder packing problem – and proposes a Simulated Annealing algorithm to solve it. The combinatorial nature of this problem does not favor the application of “exact optimization methods” and therefore the development of efficient heuristic techniques gains special attention.

Two Simulated Annealing based algorithms to solve the related problem of fitting circles inside a rectangle were introduced – the MinOSA and the MaxNSA. Although similar in nature, these two algorithms differ in the options made considering some specific aspects of

the implementation such as the objective functions and the solution spaces adopted. In the context of the implementation of these two Simulated Annealing based algorithms arose the need of proceeding to exhaustive tuning tests to adequate the parameter values of each algorithm. The results from this tuning process suggest that the algorithms are highly sensitive to the parameter values in what concerns efficiency.

The two algorithms above were used in specific cylinder packing algorithms, CPA-MinOSA and CPA-MaxNSA, which are based in MinOSA and MaxNSA, respectively (in the scope of this work, CPA-MaxNSA is coincident with the MaxNSA).

Some computational tests were performed over a varied set of test instances in order to evaluate the algorithms performance and its adequacy to solve the cylinder packing problem. The results obtained suggest that, although quite efficacious in any situation, the efficiency of the algorithms decreases as the ratio *pallet area/circle area* increases. It is also notorious the greater efficacy introduced by Simulated Annealing when compared with pure local search algorithms. From the comparison between the two cylinder packing algorithms (CPA-MinOSA and CPA-MaxNSA) stands out the greater robustness of CPA-MinOSA regarding efficacy. The existence of overlap in the solutions obtained with CPA-MaxNSA suggests the need of a post-optimization process with the purpose of guaranteeing its feasibility concerning the cylinder packing problem. The need to tune parameter β considering the relative dimensions of the pieces involved has also been identified as a possible way to improve MaxNSA performance.

The versatility of the algorithms developed is supported by the extension of the MaxNSA algorithm to the problem of packing non-identical circles in a rectangle, which was easily carried out with quite satisfactory results.

This work was partially supported by PRAXIS XXI/2/2.1/TPAR/2046/95 and PRAXIS XXI/BD/4324/96

References

- [Cor00] Correia, M.; Oliveira, José F. & Ferreira, J. Soeiro (2000). A new upper bound for the cylinder packing problem. To appear in *International Transactions in Operational Research*.
- [Dow91] Dowsland, K.A. (1991). Optimising the palletisation of cylinders in cases. *OR Spektrum*, **13**, 204-212.
- [Fra94] Fraser, H. & George, J. (1994). Integrated container loading software for pulp and paper industry. *European Journal of Operational Research*, **77**, 466-474.
- [Gar79] Garey, M. & Johnson, D. (1979). *Computers and Intractability: A guide to the theory of NP-completeness*. W.H. Freeman & Co., San Francisco.
- [Geo95] George, J.; George, J. & Lamar, B. (1995). Packing different sized circles into a rectangular container. *European Journal of Operational Research*, **84**, 693-712.
- [Gol70] Goldberg, M. (1970). The packing of equal circles in a square. *Mathematics Magazine*, 24-30, Jan-Feb 1970.

- [Hif94] Hifi, M.; Paschos, V.T. & Zissimopoulos, V. (1994). Circular cutting problem: A simulated annealing approach. Technical report 94.15, CERMSEM-Université de Paris I – Pantheon – Sorbonne.
- [Ise91] Isermann, H. (1991). Heuristiken zur Lösung des zweidimensionalen Packproblems für Rundgefäße. *OR Spektrum*, **13**, 213-223.
- [Mar95] Maranas, C.D.; Floudas, C.A. & Pardalos, P.M. (1995). New results in the packing of equal circles in a square. *Discrete Mathematics*, **142**, 287-293.
- [Pir92] Pirlot, M. (1992). General local search heuristics in combinatorial optimization: a tutorial. *Belgium Journal of Operations Research, Statistics and Computer Science*, **32**, 7-67.
- [Ree95] Reeves, C. (editor) (1995). *Modern Heuristic Techniques for Combinatorial Problems*. Advanced Topics in Computer Science. McGraw-Hill.
- [Sto98] Stoyan, Y.G. & Yaskov, G.N. (1998). Mathematical model and solution method of optimization problem of placement of rectangles and circles taking into account special constraints. *International Transactions in Operational Research*, **5**, 45-57.
- [Vid93] Vidal, R. (editor) (1993). *Applied Simulated Annealing*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag.