A System for the Compactation of Two-Dimensional Irregular Shapes based on Simulated Annealing

Valério M.M. Marques
CAPS/LRPI*

Carlos F.G. Bispo
CAPS/LRPI

João J.S. Sentieiro
CAPS/LRPI

Abstract

The problem of compacting a given number of two-dimensional shapes minimizing the area of the enclosing rectangle, i.e., minimizing the waste produced, arises quite often in some industrial processes like the automotive industry, clothing manufacturing, steel construction, electronic engineering and leather cutting.

A simulated annealing approach for the compactation of two-dimensional irregular shapes is presented. The energy function is defined by considering three components: 1) A measure of the enclosing rectangle area; 2) A measure of the distances between each piece and the center of the board, weighed by parameters reflecting the desired width/height ratio of the enclosing rectangle; 3) A measure of the quality (*goodness*) of local solutions.

The results show that the annealing algorithm performs rather well dealing with irregular patterns allocation, even though leading to higher computation times than those needed to run some heuristic methods. However, there is some evidence in the results obtained so far that near-optimal solutions may be reached in polinomial time.

1 Introduction

The problem of compacting a given number of two-dimensional regular or irregular shapes minimizing the waste produced, is certainly of relevant interest to some industries, where the cost of the wasted material can reach surprisingly high values.

To find the optimal allocation of two-dimensional irregular shapes by complete enumeration is clearly an exponencial time algorithm: given a set of N pieces to allocate over a grid of $L \times W$ points, the cardinality of the set of configurations, including the ones where overlapping occurs, is

$$|\mathcal{C}| = (L \times W \times R)^N, \quad (1)$$

where R denotes the number of possible orientations for each piece.

If the set of feasible configurations $\mathcal{F}$ is considered, by excluding the configurations with overlapped pieces, the total number of solutions is heavily dependent on the shapes of the pieces to be allocated. An upper bound for $|\mathcal{F}|$ is given by

$$|\mathcal{F}| \leq \prod_{i=0}^{N-1} \left\{ \left[(L \times W) - \sum_{j=1}^{i} A_j \right] \times R \right\}, \quad (2)$$

where A_j is the number of grid points covered by the j^{th} piece.

Since $A_j > 0$ for $j = 1...N-1$, $|\mathcal{F}| < |\mathcal{C}|$, as expected.

Unfortunately, there is no known method for the generation of $\mathcal{F}$, avoiding the search over the whole set $\mathcal{C}$.

During the second half of this century, some approaches have been tried to find good solutions for the allocation problem, partly due to worldwide industry development, and also because computers emerged as excellent tools for the solution of these problems. The methods which have been used belong basically to one of three categories:

- Operator based:

  An operator uses a CAD system to elaborate a solution.

- Algorithmic:

  In this case, solutions are found automatically by algorithms running on a computer. This problem has been adressed following two main guide lines:

*Centro de Análise e Processamento de Sinais/ Laboratório de Robótica e Processamento de Informação, Complexo I - IST, Av. Rovisco Pais, 1096 Lisboa Codex, Portugal.

 CH2976-9/91/0000-1911 $1.00

- Linear, integer and dynamic programming:
  It is specially used in applications where all the pieces have a common and regular shape-type, such as rectangles [3]. These algorithms have exponential time complexity and are heavily affected by the NP-Hard nature of the problem.
- Heuristic:
  In this approach, heuristics are used along with state space search methods to shorten the search for a solution [4]. These heuristics are however problem specific and usually based on some features of the pieces to be allocated.

- Hybrid:
  This last category is just a combination of the previous two, often allowing the draftsman to make minor rearrangements in the configuration proposed by the algorithm, or to set an initial configuration upon which the algorithm will iterate.

The method described herein belongs to the second category.

2 The Annealing Algorithm

The annealing algorithm, also called *probabilistic hill climbing* or *statistical cooling*, was first introduced by Kirkpatrick *et al.*[7] as a Monte Carlo method for combinatorial optimization. It is an adaptation of the Metropolis algorithm [9] used in the simulation of physical systems in statistical mechanics to compute average properties of the system in equilibrium at a given temperature.

Metropolis *et al.* used a simple algorithm to simulate the interactions between a finite number of atoms in equilibrium at a given temperature. In each step of the algorithm, a new state of the system was obtained by randomly selecting an atom and by randomly displacing it. The change in the energy of the system, ΔE, was then computed. If $\Delta E \leq 0$, the new state was accepted, otherwise it would be accepted with probability $\exp(-\Delta E/KT)$, where K is Boltzmann's constant, T is the temperature in Kelvin degrees and ΔE is measured in Joules.

It was shown that by repeating these steps a sufficiently high number of times, the system will evolve towards a distribution of states in which the probability of a given state s_i (with energy e_i) to be the current state is

$$\frac{\exp(-e_i/KT)}{\sum_{j=1}^{N}\exp(-e_j/KT)} \tag{3}$$

This equilibrium distribution is known as *the Boltzmann distribution.*

From (3) it is clear that for very high values of T, each state has almost equal chances of being the current one, while for very low values of T, only the states with low energies have considerable chances of prevailing.

As was stated before, these probabilities are derived for an equilibrium condition, that is, for a sufficiently high number of steps. A low temperature does not necessarily mean that the system will evolve towards a ground state, i.e., a state with minimum energy.

Annealing is so the technique developed to bring a substance into a ground state. It consists of heating the substance until a very high temperature is reached, and then to cool it slowly enough to maintain a quasi-equilibrium condition. If this is not ensured, the substance may crystallize defectively or not crystallize at all.

The major contribution of Kirkpatrick was to find that the Metropolis algorithm could be transposed to the problem of combinatorial optimization. For that purpose he replaced the energy of the system by a cost function, and reformulated Metropolis' moves as moves in the system's state space.

The annealing algorithm may be seen as a series of homogeneous Markov chains, each with a temperature lower than the previous one. Special attention must be paid to the annealing schedule. The temperature of the first chain must be high enough to allow virtually every configuration, regardless of its energy. This implies, of course, the connectivity of the state space, i.e., for every pair of states s_i and s_j belonging to the state space, there must be a finite sequence of moves with non-zero probability that connects them [1]:

$$\begin{aligned}
&\forall s_i, s_j \in \mathcal{S},\\
&\exists K_{ij} \in N, \{s_{k_1}, s_{k_2}, ..., s_{k_{K_{ij}}}\} \subset \mathcal{S}\\
&\textit{such that}\\
&s_{k_1} = s_i, s_{k_{K_{ij}}} = s_j\\
&\textit{and}\\
&P(s_{k_l}, s_{k_{l+1}}) \neq 0,\ \forall l \in \{1, ..., K_{ij} - 1\}
\end{aligned} \tag{4}$$

where $\mathcal{S}$ represents the state space and $P(s_i, s_j)$ denotes the probability that a move will take the system from state s_i to state s_j.

If temperature is decreased slowly enough from chain to chain, and the length of each chain is sufficiently large to maintain an equilibrium condition, the system will evolve with probability 1 towards one of the ground states.

There is a direct relation between the size of the decrements in temperature and the length of the Markov chains: smaller decrements allow shorter chains while larger decrements demand longer chains.

It is possible to compute an estimate of the distance to an equilibrium condition, thus calculating automatically when to decrement the temperature.

Aarts and van Laarhoven [1] propose a method to estimate the distance from an equilibrium condition.

Simulated annealing has been used in several fields where combinatorial optimization problems exist. Examples of these are the Travelling Salesman Problem [5,7], placement and routing in VLSI [7,8], neural nets [2] and image restoration [6].

3 The Proposed Approach

In the approach proposed in this paper each piece is characterized by a set of vertices, a center and an orientation. The center corresponds to the center of the smallest enveloping circumference.

At a given time instant, the state is given by the center positions and orientations of the whole set of N pieces. An energy E is defined as a weighed sum of three components, reflecting the quality (goodness) of a given configuration.

$$E = \sum_{i=1}^{3} w_i . E_i \tag{5}$$

These components are:

1. E_1 defined as the square root of the area of the smallest enclosing rectangle. The square root is used in order to ensure that all components are expressed in the adequate units.

$$E_1 = \sqrt{(\mathcal{R} - \mathcal{L}) \times (\mathcal{U} - \mathcal{D})}, \tag{6}$$

with

$$\mathcal{L} = \min_{i=1}^{N} (\min_{j=1}^{V_i} x_{ij}), \tag{7}$$

$$\mathcal{R} = \max_{i=1}^{N} (\max_{j=1}^{V_i} x_{ij}), \tag{8}$$

$$\mathcal{D} = \min_{i=1}^{N} (\min_{j=1}^{V_i} y_{ij}), \tag{9}$$

$$\mathcal{U} = \max_{i=1}^{N} (\max_{j=1}^{V_i} y_{ij}), \tag{10}$$

where V_i denotes the number of vertices of the i^{th} piece.

2. E_2 defined as the sum of the distances from each piece's farthest vertice to the board (work space) center.

$$E_2 = \sum_{i=1}^{N} d_i \tag{11}$$

with

$$d_i = \max_{i=1}^{N}((a/b).\max_{j=1}^{V_i}|x_{ij} - x_B|, \max_{j=1}^{V_i}|y_{ij} - y_B|) \tag{12}$$

where x_{ij} and y_{ij} represent the position of the j^{th} vertice of the i^{th} piece in plane XY, while x_B and y_B represent the position of the center of the board in the same plane, and a and b are weights.

This component leads to configurations in which the pieces tend to gather within a rectangle, with length/width relation equal to a/b.

3. E_3 defined as a measure of the quality (goodness) of local solutions. For each piece, the proximity between its center and its neighbours centers is computed, contributing nonpositively to the total energy:

$$E_3 = \sum_{i=1}^{N} \sum_{j \in \nu(i)} (c_{ij} - R_i - R_j). \tag{13}$$

$\nu(i)$ denotes the set of indexes corresponding to pieces which are neighbours of the i_{th} piece. Two pieces are said to be neighbours if their enveloping circumferences overlap. R_i represents the radius of the enveloping circumference of the i_{th} piece.

c_{ij} is the euclidian distance between the centers of the i^{th} and j^{th} pieces.

E_3 is therefore nonpositive, rewarding good local arrangements.

A move consists of a translation, a rotation or, in case where the pattern has no symmetry axis, of an inversion of a pattern. This move may or may not lead to a legal configuration. Legal configurations are the ones in which no overlapping between pieces occurs. Illegal configurations are always rejected. This fact has an important effect on the total number of configurations given by (1), since only legal configurations are allowed, but also affects the topology

of the state space, since any path linking two legal states s_i and s_j including at least one illegal state is now excluded. This can be seen in (4), where it may no longer be true that $P(s_{k_l}, s_{k_{l+1}}) \neq 0 \ \forall l \in \{1, ...K_{ij} - 1\}$, since $s_{k_{l+1}}$ may now be an illegal state.

If, on the other hand, a move leads to a legal configuration, the resulting change in the energy is measured. In case there is a decrement in E, the new configuration is accepted, otherwise the probability of its acceptance will be given by

$$P(\Delta E, T) = \exp(-\Delta E/T) \tag{14}$$

T being the current temperature.

The NP-Hard nature of this problem shows its effects even for low values of N, thus making impossible an exhaustive or even vast search of the state space in a reasonable amount of time. A direct consequence of this is that one cannot expect the algorithm to run under quasi-equilibrium conditions, at least for a large set of pieces and/or an allocation grid with many points. This implies that the initial configuration may have significant importance in the final solution and that the probability that the system will evolve towards a ground state is now significantly less than 1.

4 The Annealing Schedule

As was stated before, the behaviour of the annealing algorithm depends heavily on the proximity of the system from equilibrium. It is therefore crucial that the temperature evolution takes this factor into account.

Four parameters are responsible for the temperature evolution:

- The initial value of the temperature;

  This value is obtained as a function of the maximum difference in cost between any two neighbour configurations.

$$T_0 = k \times \max_{i,j} \Delta C_{ij}, \ k >> 1,$$

  with $i = 1...|\mathcal{F}|, j \in \nu(i)$.

- The decrement function of the temperature;

  Exponential cooling was chosen:

$$T_n = f \times T_{n-1}, \tag{15}$$

  with f usually taking values over [0.80;1.0[.

- The length of each Markov chain;

  As was stated before, this length is related to the decrement function of the temperature.

- The stopping criterion.

  An adequate stopping criterion consists on terminating the algorithm when a given number of Markov chains results in no further enhancement of the best solution yet found.

5 Some implementation details

As was mentioned in the previous section, special attention should be given to the definition of the initial configuration. Several strategies for the generation of the initial distribution are under consideration. In the work described in this paper, an automatic generation strategy was adopted. Other options will be considered in the future.

The fact that after every move it is necessary to verify the existence of overlapping between the moved piece and the others, leads to an increase in the processing time, specially when there are many sides to each piece. In order to reduce the computation time of this routine, each piece was involved in an enveloping circumference, just wide enough to include its vertices. When the overlapping test is performed, the routine will check the overlapping between the correspondent enveloping circumferences, which is a fast operation. Only if this test is positive, the routine will verify the overlapping between all the sides of both pieces. This two-phase procedure permits the saving of an important amount of time.

When the pieces are first designed,the enveloping circumferences are determined automatically, and the existence of symmetry axis is checked. This is necessary to determine the need for inversions. Only shapes without symmetry axis need to be inverted in order to ensure all possible configurations.

During the experiments described in the next section, every piece's rotation was restricted to multiples of 90 degrees, therefore limiting the total number of legal configurations. As expected from (2), this fact is of great significance to the state space dimension.

The algorithm was programmed in C language running on a DEC 3100 Risc Station.

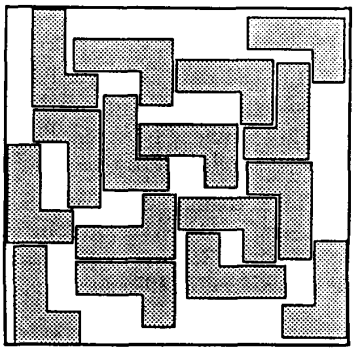

Figure 1: Layout example for 16 L-shaped patterns by iterative improvement.

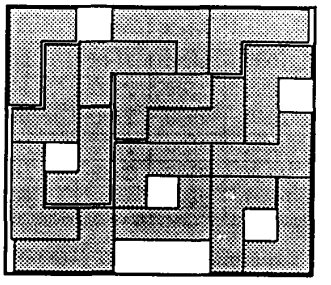

Figure 2: Layout example for 16 L-shaped patterns with quick schedule.

6 Experimental Results

Exhaustive results are not yet available. However, the results obtained so far are significant enough to allow claiming the feasibility of the approach.

Some typical results obtained with the annealing algorithm previously described are now presented.

Figures 1, 2 and 3 show the final allocations for a set of 16 L-shaped patterns, obtained respectively with a greedy schedule, equivalent to a search by iterative improvement, a quick schedule and a slow schedule. Square configurations were pretended by setting $a = b$ in (12). These figures show, as expected, better results for longer schedules. Annealing too quickly often produces "jammed" configurations.

Table 1 shows the results for 6 combinations of f (see equation (15)) and L, the length of each Markov chain. These experiments were done with initial temperature $T_0 = 20.0$. The reported time refers to the average computing time for one single run.

Figure 4 show a layout for a more realistic set, typical in clothing manufacturing. The set is similar to one presented in [4].

Figure 4 was obtained from a series of 10 runs of the annealing algorithm, described as experiment 6 in table 1.

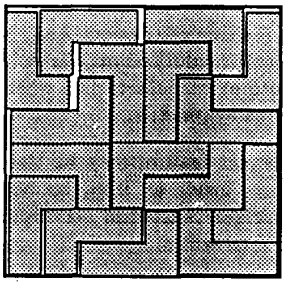

Figure 3: Layout example for 16 L-shaped patterns with slower schedule.

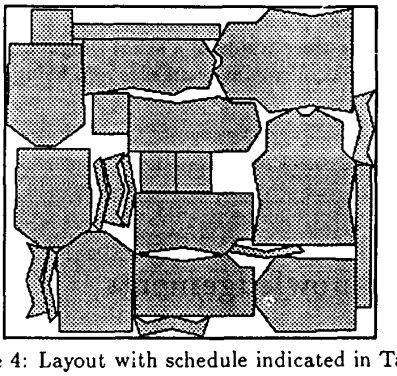

Figure 4: Layout with schedule indicated in Table 1, Experiment 6.

7 Conclusions and Future Steps

Although the results are still few, some preliminary conclusions may already be drawn.

The annealing algorithm proves to be an adequate tool to solve the allocation problem, as formulated.

Exp.	f	L	Min. Area	Avg. Area	Computing Time (s)
1	0.90	1000	2834	3148	2029
2	0.95	1000	2609	2899	2797
3	0.99	1000	2530	2700	9574
4	0.90	10000	2586	2706	14417
5	0.95	10000	2522	2680	21437
6	0.99	10000	2481	2560	88986

Table 1: Experimental results (see text).

Though the computation time may, in same applications, be considered high, this method appears to achieve near-optimal solutions in polinomial time.

Extended capabilities are under study. For instance, the inclusion of defective areas inside the board may be a realistic restriction. Also, in some applications, symmetric solutions may be welcome, as it is the case in clothing manufacturing, where patterns in a shirt are often symmetric.

A more important feature may be the ability to allocate the pieces inside an irregular board. This allows reusing surfaces wasted in previous cuts, thus diminishing the final waste.

This last point is also important because, once implemented, it allows constructions where pieces with holes are first filled with smaller pieces, by considering these holes also as irregular boards and using the annealing algorithm to allocate these smaller pieces inside it. Only then the allocation inside the whole board will take place. For that purpose, a new system, allowing the packing of a set of irregular shapes inside irregular, non-convex areas is now being developed.

8 Acknowledgements

The authors would like to acknowledge Mário Figueiredo, José Bioucas and Luís Custódio for some useful discussions on annealing and the compactation problem and to Christopher and Ana Bodin, for helping preparing this document.

References

[1] Aarts, E.H.L. and van Laarhoven, P.J.M., "Statistical Cooling: A General Approach to Combinatorial Optimization Problems," Philips Journal of Research, vol. 40, No. 4, pp.193-226, 1985.

[2] Ackley, D.H., Hinton, G.E., and Sejnowsky, T.J., "A Learning Algorithm for Boltzmann Machines," Cognitive Science: A Multidisciplinary Journal of Artificial Intelligence, 9, pp.147-169, 1985.

[3] Adamowicz, M. and Albano, A., "A Solution of the Rectangular Cutting-Stock Problem," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-6, no. 4, pp. 302-310, April 1976.

[4] Albano, A. and Sapuppo, G., "Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Methods," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, no. 5, pp. 242-248, May 1980.

[5] Černy, V., "Thermodinamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm," Journal of Optimization Theory and Applications, 45, pp. 41-51, 1985.

[6] Geman, S. and Geman, D., "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, 6: pp. 721-741, 1984.

[7] Kirkpatrick, S. , Gelatt Jr., C.D. and Vecchi, M.P., "Optimization by Simulated Annealing," Science, 220, pp. 671-680, 1983.

[8] Mallela, S. and Grover, L.K., "Clustering Based Simulated Annealing for Standard Cell Placement," In *Proceedings of the 25th ACM/IEEE Design Automation Conference*, pp. 312-317, 1988.

[9] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E., "Equation of State Calculations by Fast Computing Machines," Journal of Chemical Physics, 21, pp. 1087-1092, 1953.