Application of simulated annealing in improving the performance of stereolithography

V.Canellidis, V.Dedoussis, S.Sofianopoulou

Center for Product Development and Rapid Prototyping, University of Piraeus, Piraeus, Greece

ABSTRACT: Effective utilization of Stereolithography (SL) mainly relies on orienting and packing parts optimally on the fabrication platform of the machine, so to achieve maximum space utilization and minimum build time, without of course compromising surface quality. The present work focuses on an effective way to pack parts optimally on the fabrication platform of SL machine. Due to technical constrains set by SL technology, the original 3-D packing problem is simplified by one dimension by projecting each one of the parts on the build platform (*x-y* plane) and packing their projections instead of the actual parts themselves. In order to solve the resulting 2-D packing problem a heuristic method has been adopted. The heuristic method consists of a Simulated Annealing algorithm employing a polynomial-time cooling schedule and a new improved placement rule.

1 INTRODUCTION

Stereolithography is one the most widely used techniques belonging to the group of Layer Manufacturing Technologies (L.M.T) or else Rapid Prototyping Technologies (R.P.T). Rapid prototyping technologies involve fabrication of parts directly from 3D-CAD models by successive addition of layers of various kinds of materials like polymers, metals, ceramics or paper. In stereolithography parts are fabricated by successive solidification (polymerization) of thin resin layers. For the solidification of the layers an UV laser beam, focused on the surface of the resin, is used. The laser beam, guided by a system of galvanometer mirrors, scans selected areas at the surface of the resin, thereby solidifying thin resin layers. Next, the platform, on which the part is attached, lowers a small distance (0.10-0.25 mm) and a recoating blade (sweeper) moves across the resin surface applying and leveling a new layer of liquid resin on top of the already fabricated part of the part. The process continues in the same manner, gradually building the part from the bottom-up, until the entire part is fabricated.

In the past few years researchers have investigated extensively the time needed for a part to be built by stereolithography systems (e.g. Giannatsis et al. 2001). Thus the build-time has been expressed as:

$$\sum_{i=1}^{n} TLayer(i) \qquad (1)$$

where *TLayer* is the time required for the addition of the ith layer and n is the total number of layers. The addition of a layer consists of recoating and scanning. Thus the time needed for the fabrication of a single layer may be expressed as the sum of recoating and scanning time. The recoating time now can be calculated as the sum of the time required for the lowering of the platform, the time required for the recoater movement and the pre- and post-scan delay periods.

The time required for the lowering of the platform, the time required for the recoater movement and the pre- and post-scan delay periods have been found to be independent of the number of parts being fabricated at a time. So, if we proceed in fabricating one part at a time instead of fabricating as many parts as possible, the total sum of build-time will be significantly larger. Noting that the total cost of ownership (TCO) increases as the build time increases, it becomes obvious that effective utilization of stereolithography relies on packing parts optimally on the fabrication platform of the machine.

Quite a few researchers have examined the problem of packing several parts on the fabrication platform of the machine. The methodologies and algorithms employed tackle the relevant 3D bin packing problem mainly for the case of Selective Laser Sintering technology. In most of the cases, heuristic methods have been used in conjunction with a placement rule (e.g. Hur et al. 2001) to produce near optimal solutions. The corresponding work in stereolithography technology, however, is rather limited. Decisions regarding packing rely mainly on the skill and experience of the stereolithography apparatus operator. Thus the purpose of this work is to present a methodology that "automates" the process of making these decisions for the stereolithography technology and provides the operator with effective packing solutions.

Due to the existence of support structures in SL technology, packing solutions/fabrication layouts where some parts are built on top of others are unacceptable. Thus, the original 3-D packing problem is simplified by one dimension, by projecting each one of the parts on the build platform (*x-y* plane) and packing their projections instead of the actual parts themselves. The parts are being projected on the build platform only after the calculation of the desirable orientation for each part. Then the minimum bounding rectangle is being defined for each one of the projections. Finally the minimum bounding rectangles are being used as input in the packing algorithm.

The packing algorithm consists of a Simulated Annealing procedure, which utilizes a polynomial time cooling schedule, in conjunction with a new improved placement rule.

2 SIMULATED ANNEALING OVERVIEW

Simulated annealing is a generalization of the Monte Carlo method, suitable for large scale optimization problems, especially those where a desired global extremum is hidden among many, poorer, local extrema. It was motivated by an analogy to the thermodynamics of annealing in solids. In an annealing process, material is being heated to a temperature that permits many molecules to move freely with respect to each other. Then it is cooled in a slow manner, until the material freezes into a crystal, which is completely ordered, and thus the system is at the state of minimum energy. The temperature of the material must be gradually lowered so that at each temperature the atoms can move enough to begin adopting the most stable orientation. Noting that the target in a combinatorial optimization problem is to find an optimal solution or else the "minimum energy state of the problem", simulated annealing technique uses an analogous cooling operation for transforming a poor, unordered solution into an ordered, desirable solution, so as to optimise the objective function.

By analogy to the physical annealing, simulated annealing uses the control parameters that follow:

Cost function: An objective function that measures how well the system performs when a certain configuration is given.

Move schemes: A generator of random changes in the configuration so as to create new candidate solutions.

Temperature: A control parameter analogous to the temperature in the annealing process of solids. This artificial parameter acts as a source of stochasticity, which is convenient for eventually detrapping from local minima, and it represents the willingness of a system to accept a state/solution that is worse than the current.

Cooling schedule: A definition of the cooling speed to anneal the problem from a random solution to a good, frozen one. In its details, it must provide a starting temperature, together with the rules to determine when and how much the temperature should be reduced and when annealing should be terminated.

Metropolis acceptance criterion: As it was pointed out simulated annealing is suitable for problems where the desired global optimum is hidden among many, poorer, local optima. Thus the method should be capable of escaping local optima. In order to achieve the previous mentioned goal, simulated annealing takes not only downhill moves, but also permits uphill moves with an assigned probability of $P(\Delta C) = \exp(-\Delta C/T)$ where ΔC is the change in the objection function value and T is the temperature parameter. Thus a new candidate solution is accepted, if it has the same or smaller objective function value. However, if the new solution yields an increase in cost of the system, then the new solution is judged for suitability probabilistically according to the probability $P(\Delta C) = \exp(-\Delta C/T)$. Thus the new solution is being accepted if a random number, generated in the interval (0,1), is less than or equal to $P(\Delta C) = \exp(-\Delta C/T)$. Otherwise, it is rejected.

A simulated annealing optimization starts with a Metropolis Monte Carlo simulation at an initial high temperature T. This means that a relatively large percentage of the random steps that result in an increase of the cost of the objection function will be accepted with probability $P(\Delta C) = \exp(-\Delta C/T)$. Metropolis Monte Carlo simulation is a randomization technique being used for optimizing a function using a random sampling of the solution space. After a sufficient number of Monte Carlo steps, the temperature is decreased. The Metropolis Monte Carlo simulation is then continued. This process is repeated until the final temperature is reached. Actually a simulated annealing algorithm consists of a pair of nested loops. The outer loop controls the temperature parameter and the inner loop (or else Markov chain) runs a Metropolis Monte Carlo simulation at that temperature. The way in which the temperature is decreased has already been referred as the *cooling schedule*.

3 PROBLEM FORMULATION

In this section, the basic concepts of the methodology utilized for solving the packing problem are described in detail. As it was mentioned before, our target is to pack as many parts as possible on the platform of a stereolithography system in order to reduce the TCO or else to minimize the unoccupied area (trim loss) on the platform. Thus, we are dealing with a minimization problem and the cost function F may be defined as the percentage of the area of the platform that is unused by the, say *n*, rectangles:

$$F=1-\frac{\sum_{i}^{i=n} l_i w_i}{lw} \qquad (2)$$

where *l* and *w* are the length and the width of the fabrication platform respectively, and l_i, w_i are the length and width of the *i*th rectangle.

A packing pattern can be represented by a permutation $\pi=(i_1,\dots..i_n)$ where i is the index of the r_i rectangle. Actually, a permutation represents only the sequence in which the rectangles will be packed by a placement rule. The advantage of this data structure is the facile creation of new permutation-solution by simply changing the sequence. For the assignment of each permutation to an unique packing pattern a placement rule is being utilized which will be discussed later in detail. After the creation of the packing pattern we are able to calculate its "cost" through the cost function.

3.1 *Cooling schedule*

The cooling schedule is being considered to be the most important factor in a simulated annealing algorithm. It is composed by the starting temperature and the rules to determine when and how much the temperature should be reduced and when annealing should be terminated. The cooling schedules may be divided into two broad groups, static and adaptive. Static schedules are those schedules that follow a predetermined course of decrement. Static schedules generally disregard the dynamic behaviour of the problem at hand. On the other hand, dynamic schedules use statistical analysis of objective function values examined to control the temperature decrement, while they are computational expensive.

It is apparent that if the temperature parameter is kept high or it is decreased too quickly then the process will not tend toward globally optimal configurations. Therefore, an efficient cooling schedule must be designed in order to successfully implement the simulated annealing algorithm. Thus, a dynamic polynomial-time cooling schedule has been adopted as the most promising approach.

By using the polynomial-time cooling schedule suggested by Van Laarhoven (1988), the initial temperature is being determined by generating candidate configurations/solutions and evaluating their suitability according to the Metropolis Monte Carlo algorithm. At the beginning of the annealing process the acceptance rate of new configurations, disregarding the objection function values, should be high enough in order not to get trapped easily in a local minimum. The acceptance rate may be expressed as:

$$x \cong \frac{m_1 + m_2\exp\left(\frac{-\overline{\Delta C^+}}{T}\right)}{m_1 + m_2} \qquad (3)$$

in Eq. (3), C is defined as the cost of the objection function, m_1 as the number of transitions occurred resulting in decrease of the objection function ($\Delta C \leq 0$), m_2 as the number of the transitions occurred resulting in increase of the objection function ($\Delta C>0$) and finally $\overline{\Delta C^+}$ is defined as the average increase in cost over m_2 transitions. Eq. (3) can now be rewritten as:

$$T=\frac{\overline{\Delta C^+}}{\ln\left(\frac{m_2}{m_2\cdot x - m_1\cdot(1-x)}\right)} \qquad (4)$$

By generating a fixed number of transitions and accepting all configurations with increased objective function values an initial temperature can be obtain using Eq.(4).

After determining the initial temperature a decrement rule must be established. Keeping a record of the cost values of the configurations $\pi_1,\dots,\pi_j$ that occur during the generation of the *k*th Markov Chain (the inner loop), where j is the length of the *k*th Markov Chain, we are able to approximate the probability distribution of the cost values of the *k*th Markov Chain by a normal distribution with mean μ_k given by

$$\mu_k=\frac{1}{j}\sum_{i=1}^{i=j} C(\pi_i) \qquad (5)$$

and variance

$$\sigma_K^2=\frac{1}{j}\sum_{i=1}^{i=j} C^2(\pi_i)-\mu_K^2 \qquad (6)$$

Therefore, the decrement rule for the temperature may be expressed as

$$T_{k+1}=\frac{T_k}{1+\frac{T_k\cdot\ln(1+\delta)}{3\sigma_\kappa}} \qquad (7)$$

where δ is called the distance parameter. The choice of δ determines how closely the algorithm will approximate a globally minimum state. Moreover, δ controls the computational effort that is going to be needed to reach an approximate globally minimum configuration. Finally the simulated annealing algorithm ends when temperature reaches zero.

3.2 *Move schemes*

Move schemes in the annealing process denote the generation of candidate configurations/solutions for the system. A new configuration/solution may or may not be accepted as the next state of the system depending upon the Metropolis criterion. Typically, modifying the current state of a system in some way is creating a new solution. In the presented methodology three move schemes have been adopted. The first one simply selects two rectangles from a permutation $\pi = (i_1,\ldots..i_n)$ and swaps their locations. For instance say that the current state of the system corresponds to the permutation $\pi = (1, 2, 3, 4, 5, 6, 7)$ where each integer is the index of a rectangle. This move scheme would randomly select two indices, e.g. 3 and 5, and swap their locations. Thus, the new solution would be $\pi = (1, 2, 5, 4, 3, 6, 7)$.

The second move scheme was inspired by the SJX operator, introduced by Jakobs (1996) as a crossover operator in genetic algorithms. Given a permutation $\pi= (i_1,\ldots..i_n)$, as well as two integers *b* and *c*, the scheme takes *c* consecutive indices starting from the *b*-th index of the permutation and place them at the beginning of the new permutation. Then it fills the remaining unallocated slots of the new permutation by the remaining indices. Suppose, for example, that the initial configuration is $\pi_1 = \{1, 2, 3, 4, 5, 6, 7\}$, $b = 2$ and $c = 3$. Then 2, 3, 4 are the first three indices of the new configuration, and the other unused indices 1,5,6,7 will fill the rest of the new configuration. Thus the new solution is $\pi_2 = \{2, 3, 4, 1, 5, 6, 7\}$.

The third move scheme randomly selects the 40% of the indices of a permutation $\pi = (i_1,\ldots..i_n)$ and rotates by 90 degrees the rectangles that are being represented by those indices.

3.3 *Placement rule*

As it was mentioned before, a permutation represents only the sequence in which the rectangles will be packed. Thus, a placement rule is needed in order to perform the actual packing. Taking into account the nature of our problem, we can consider it as a 2-D bin packing problem, which is commonly encountered in the field of operational research. In order to solve this kind of problems the majority of the researchers use the standard bottom-left (BL) placement policy which attempts to minimize the total length required by placing each part's bounding rectangle (BR) as far to the left as possible and by favouring positions near the bottom-lower and side of the container/platform. The disadvantage of standard bottom-left algorithm is that groups of polygons exist for which an optimal packing pattern cannot be generated. Thus, a new improved placement rule should be established in order to overcome the previously mentioned drawbacks.

Before describing the algorithm the terms vertical insertion lines and horizontal insertion lines should be explained. Vertical insertion lines correspond to the lines, normal to the lower side of the platform, which are adjacent to the rightmost side of each rectangle that has been placed on the fabrication platform. Horizontal insertion lines are the lowest, in the *y*-wise sense, lines parallel to the *x*-axis of the platform that 'prevent' the overlapping of the rectangle to be packed with the rectangles already placed on the platform. Where a vertical and a horizontal insertion line intersect there is an insertion point where a new rectangle can be placed without overlapping with the rectangles already placed on the platform. Thus, there are as many insertion points as vertical insertion lines.

For clarity reasons, the algorithm of placing-packing the BR on the SLA fabrication platform will be described with reference to Figures 1-5. The algorithm starts by placing the first BR on the platform employing a simple BL rule as it is shown in Figure 1.

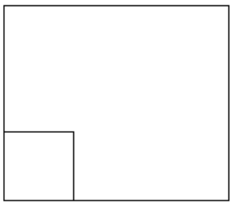

Figure 1. Placement of first rectangle

After the placement of the first rectangle there are two vertical insertion lines in order to pack the second rectangle. The first vertical insertion line coincides with the left side of the platform, while the second is adjacent to the rightmost side of the rectangle that has already been packed. Moreover, there are two horizontal insertion lines respectively to the vertical insertion lines. A horizontal insertion line that is adjacent to the upper side (in the y-wise sense) of the rectangle already packed corresponds to the first vertical line. A horizontal line that coincides with the lowest side of the fabrication platform corresponds to the second vertical line. As it is obvious, a horizontal line is being used simply as a mean to prevent overlaps. Thus, there are two insertion points where the second rectangle may be placed. In Figures 2 and 3 one can observe the two insertion points with their vertical and horizontal insertion lines as well as the actual packing of the second rectangle in both positions. The hatched rectangle is the rectangle that we are trying to pack. The algorithm chooses that insertion point that corresponds to the lowest horizontal insertion line.

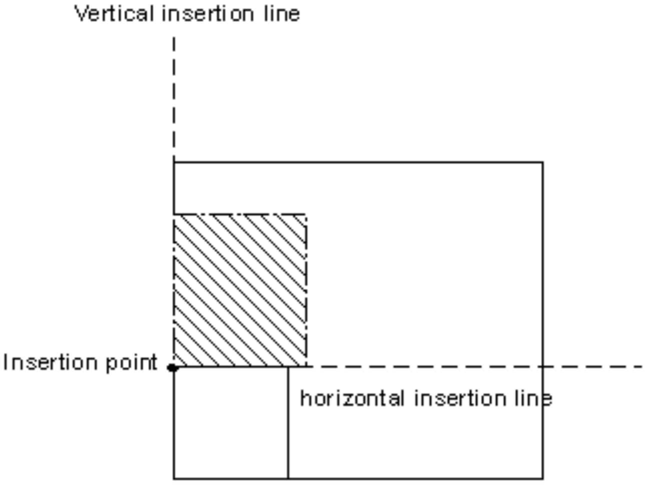

Figure 2. First possible placement of second object

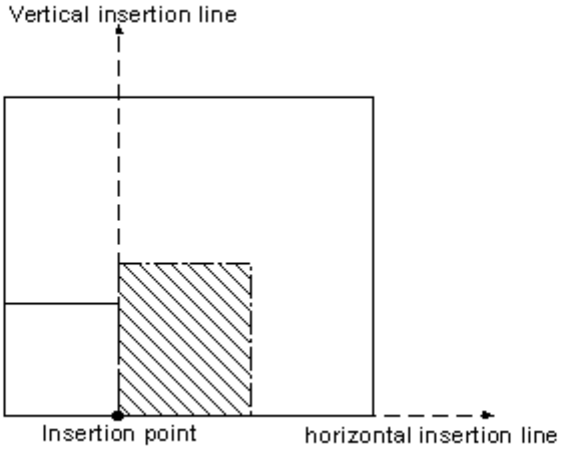

Figure 3. Second possible placement of second object

In that way the second and the third rectangle are placed. For the packing of the fourth rectangle there are four vertical and horizontal insertion lines indicating four insertion points. The first vertical insertion line coincides with the left side of the platform. Now the horizontal insertion line for this vertical line depends on the length, in the x-wise sense, of the rectangle to be placed. For instance if the length of the rectangle to be placed is smaller than the length of the first rectangle then the horizontal insertion line is adjacent to the upper size (in the y-wise sense) of the first rectangle, as it is shown in Figure 4. If the length of the rectangle to be placed is bigger than the length of the first rectangle then the horizontal insertion line is set at the top of an already packed BR for which there would be no overlaps (e.g. Fig 5). Using the same methodology the other three vertical and horizontal insertion lines are being calculated resulting in four insertion points. Finally, the rectangle is being placed at that insertion point that owns the lowest horizontal insertion line. If an insertion point results in a layout that is unacceptable, e.g the part is packed partially outside the platform, then the algorithm simply disregards it. Finally, if two or more insertion points share a common horizontal insertion line, then the algorithm selects the one that owns a vertical line nearest to the left side of the platform.

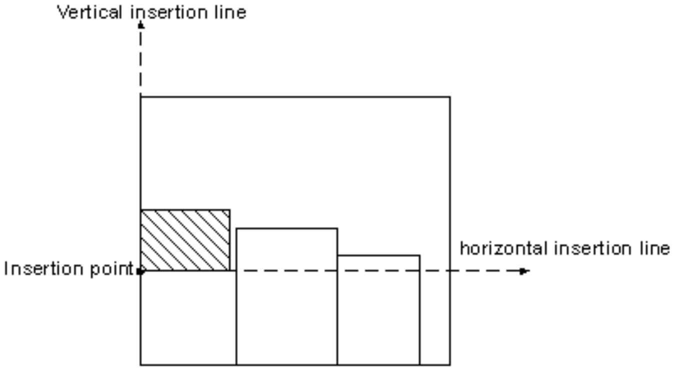

Figure 4. First possible insertion point, given a vertical insertion line

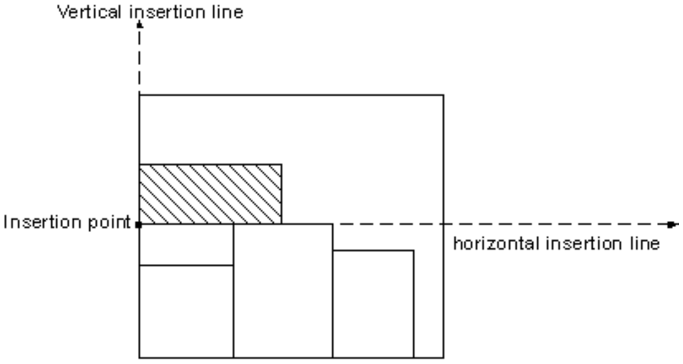

Figure 5.Second insertion point, given a vertical insertion line

3.4 *Comparison with a genetic algorithm*

The proposed methodology has been coded in visual C++ allowing the user to define the length of the internal Markov chain, the initial temperature (in case there is a disagreement with the initial temperature provided by the algorithm) and the distance parameter. Moreover, the user can define the initial acceptance percentage and the length of the trial run in order to obtain an estimate of a good initial temperature. Computational results were then compared with those produced by a methodology proposed by Mantzouratos et al. (2002) which utilizes a genetic algorithm and an improved BL placement rule in order to solve the 2-D bin packing problem. Genetic algorithms are also heuristic algorithms, which are based on the principles of natural selection and survival of the fittest. A genetic algorithm attempts to evolve a solution using a population of potential solutions, appropriately called chromosomes. New chromosomes/solutions are created through a process of breeding, where promising genetic material is passed from one chromosome to the other.

In order to compare the two heuristic procedures, four test problems proposed by Lai & Chan (1997) and Jakobs (1996) were used, as well as two others of our own. For each test problem the simulated annealing algorithm and the genetic algorithm were run 15 times and the average values where taken into account. Only the four test problems proposed by Lai & Chan (1997) and Jakobs (1996) have a known optimal solution of zero trim losses. Table 1 presents the dimensions of the fabrication platform as well as the number of rectangles to be packed for the six test problems. Finally in Table 2 the trim losses for each test and each algorithm are presented.

Table 1. Test problems

Size of fabrication platform		Number of rectangles	Known optimal solution
Width	Height		
400	200	10	Yes
400	400	15	Yes
70	80	20	Yes
70	80	25	Yes
50	50	25	No
50	50	30	No

Table 2. Trim losses per test problem and algorithm.

Number of rectangles	Known optimal solution	Trim losses by simulated annealing algorithm	Trim losses by genetic algorithm
		% of the platform area	% of the platform area
10	Yes	0	0.045
15	Yes	1.25	1.25
20	Yes	4.0179	3.571426
25	Yes	3.2143	10.714287
25	No	0	0
30	No	4.24	7.49

Though simulated annealing algorithm took much more time than the genetic algorithm to converge in a “good” solution, we observe that in most cases performed very satisfactory indicating the efficiency of the proposed methodology. Moreover, it must be pointed out that part of the success of the methodology is due to the use of a new improved placement rule which results in much more dense packing layouts than the one employed by the genetic algorithm.

4 CONCLUSIONS

In this paper, a methodology that “automates” the process of making decisions regarding packing for the stereolithography technology and provides the operator with effective packing solutions was presented. The methodology consists of a simulated annealing algorithm and new improved placement rule. A software tool has been developed that can be used by the SLA operator at a pre-processing phase. Finally, the method has been found to be quite robust through the comparison with packing solutions provided by a genetic algorithm.

ACKNOWLEDGEMENT

The financial support of the University of Piraeus Research Center is gratefully acknowledged.

REFERENCE

Dowsland, K.A. & Dowsland, W.B 1995. Solution approaches to irregular nesting problems. European Journal of Operational Research 84: 506-521

Giannatsis, J., Dedoussis V, Laios L. 2001 A study of the build time estimation problem for stereolithography systems. Robotics & Computer Integrated Manufacturing 17: 295-304.

Hur, S.M, Choi, K.H, Lee, S.H, Chang, P.K. 2001. Determination of fabricating orientation and packing in SLS process, Journal of Materials Processing Technology 112:236-243.

Hopper, E., Turton, BCH. 2001. An empirical investigation of metaheuristic and heuristic algorithms for a 2D packing problem. European Journal of Operational Research 128:34-57.

Jakobs, S. 1996. On genetic algorithms for the packing of polygons, European Journal of Operational Research 88: 165-181.

Lai, K.K & Chan, W.M. 1997. Developinga simulated annealing algorithm for the cutting stock problem. Computers and industrial engineering 32(1): 115-127.

Leung, T.W, Chan. Chin Kin, Troutt, M.D. 2002. Application of a mixed simulated annealing-genetic algorithm heuristic for the two dimensional orthogonal packing problem, European Journal of Operational Research 145: 530-542.

Mantzouratos, N., Canellidis, V., Zouzoulis, T., Dedoussis, V. & Sofianopoulou, S. 2002. Pre-processing software tool for optimizing stereolithography apparatus build performance, Proceedings Euro-uRapid 2002, International Users Conference on Rapid Prototyping & Rapid Tooling & Rapid Manufacturing, editor R. Meyer, Frankfurt, Dec. 2002, paper B-5/1.

Theodoracatos, V.E. & Grimsley, J.L. 1994. The optimal packing of arbitrarily shaped polygons using simulated annealing and polynomial time cooling schedules, Computers methods in applied mechanics and engineering 125: 53-70.

Van Laarhoven, P.J.M. 1988. Theoretical and computational aspects of simulated annealing, CWI Tracts, Amsterdam.

Tian-Peng, Ma-Jian, Zhang-Dong-Mo. 1998. Application of simulated annealing to the compinatorial optimization proble with permutation property: An investigation of the generation mechanism, European Journal of Operational Research 118: 81-9