ELSEVIER

European Journal of Operational Research 93 (1996) 522–535

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

# A simulated annealing heuristic for the one-dimensional cutting stock problem

Chuen-Lung S. Chen *, Stephen M. Hart, Wai Mui Tham

*Department of Industrial Engineering, Mississippi State University, P.O. Drawer U, Mississippi State, MS 39762, USA*

Received 13 December 1993; revised 21 March 1995

**Abstract**

This paper presents a new simulated annealing approach to the solution of an integer linear programming formulation of the one-dimensional cutting stock problem. Design and implementation issues are discussed – including a thorough statistical analysis of the effects of various parameters on the efficiency and accuracy of solutions. The performance of the new algorithm is compared to that obtained using an existing simulated annealing based methodology, and results presented herein indicate that the new approach consistently generates more efficient solutions with respect to objective value and execution time.

*Keywords:* Cutting stock problem; Simulated annealing; Integer linear programming; Optimization

## 1. Introduction and background

Simulated annealing is a computational process which attempts to solve hard combinatorial optimization problems through controlled randomization. The procedure was popularized by Kirkpatrick et al. [13], and is based on work by Metropolis et al. [15] (the so-called Metropolis algorithm) in statistical mechanics. Simulated annealing emulates the physical process of annealing (hence the name of the heuristic) which attempts to force a system to its lowest energy state through controlled cooling. In general, the annealing process involves the following steps:

1. The temperature of the system is raised to a sufficient level.
2. The temperature of the system is maintained at this level for a prescribed amount of time.
3. The system is allowed to cool under controlled conditions until the desired energy state is attained.

The initial temperature (Step 1), the time the system remains at this temperature (Step 2), and the rate at which the system is cooled (Step 3) are referred to as the *annealing schedule*. If the system is allowed to cool too fast, it may "freeze" at an undesirable, high energy state. With respect to optimization problems, the state of the system corresponds to the value of the objective function. Similarly, the freezing of a system at an undesirable energy state corresponds to an optimization problem which is "frozen" at a local optimum. Given this, in simulated annealing the problem starts at some suboptimal solution, and a series of moves (changes of values of decision variables) are made according to a user-defined annealing schedule until either the optimal solution is attained or the problem becomes frozen at a local optimum from which it cannot

* Corresponding author.


*SSDI* 0377-2217(95)00102-6

improve. To avoid freezing at a local optimum, the algorithm moves slowly (with respect to the objective value) through the solution space. This controlled improvement of the objective value is accomplished by accepting non-improving moves (i.e., those which yield an objective value greater than or equal to the last accepted objective value) with a certain probability (based on the resulting change in the objective value and the current temperature) which decreases as the algorithm progresses. (The algorithm's transitions can be modeled as a collection of finite-length Markov chains corresponding to each temperature level of the system. Hence, through selection of an appropriate probability distribution and through control of its parameters, the algorithm's rate of convergence is controlled. See Van Laarhoven and Aarts [17] for a detailed mathematical description.)

The general procedure for implementing a simulated annealing algorithm follows:

*Step 1.* Select an initial temperature, $t$, and an initial solution, $x_0$. Let $f_0 = f(x_0)$ denote the corresponding objective value. Set $i = 0$ and go to Step 2.

*Step 2.* Set $i = i + 1$. Randomly generate a new solution, $x_i$, and evaluate $f_i = (x_i)$.

*Step 3.* If $f_i < f_{i-1}$, then go to Step 5. Otherwise, accept $f_i$ as the new solution with probability

$$e^{|f_i - f_{i-1}|/t}$$

*Step 4.* If $f_i$ was rejected as the new solution in Step 3, set $f_i = f_{i-1}$. Go to Step 5.

*Step 5.* If satisfied with the current objective value, $f_i$, stop. Otherwise, adjust the temperature, $t$, according to the annealing schedule and go to Step 2.

Many researchers have reported encouraging results from the application of simulated annealing to the solution of computationally complex problems. A review of these applications can be found, for example, in Eglese [3]. Finally, although simulated annealing is typically considered a heuristic procedure, sufficient conditions for asymptotic convergence of simulated annealing to the global optimum solution have been established (Lundy and Mees [14], Hajeck [9]).

The trim problem or one-dimensional cutting-stock problem arises in a variety of industries which cut rolls of raw materials to rolls of various smaller sizes according to customer demand.

A cutting machine consists of a set of knives which can be adjusted to many combinations of different locations or sizes of cuts. All knives cut through the raw material roll simultaneously resulting in smaller final rolls of different sizes per single cut. A combination of different locations of knives is termed a *cutting pattern* or a *set-up*.

The trim problem is complicated when there are multiple stocks of different sizes of raw materials and various cutting machines with different cutting efficiencies – each handling different raw material sizes. Also, a set-up cost is incurred whenever there is a change from one cutting pattern to another. This set-up cost is attributed to labor costs and loss of production time associated with the process of setting up knives, adjustment of the knives to various locations, and fine tuning to precise locations.

The objective of the trim problem is to determine the set of cutting patterns and assignment of said patterns to machines such that the sum of the trim loss costs and set-up costs are minimized. This objective is subject to imposed constraints such as fulfilling customer orders, not exceeding stocking limits of final rolls, balancing the load between machines, etc.

The trim problem can be modeled and solved as an ILP; however, the formulation typically becomes intractable when the following occur:

1. There are a substantial number of different raw material sizes. This is not unrealistic and is sometimes desirable since it could increase the choices of cutting patterns with lower trim loss.
2. There are many different final roll sizes in customer orders. This situation is especially likely to happen when final roll size is allowed to vary by regular increments (say 1 inch or 0.5 inch).

The first known formulation of a cutting-stock problem (CSP) was provided by the Russian economist Kantorovich (see Haessler and Sweeney [8]). Eisemann formulated a practical, multistock, multi-machine, one-dimensional CSP and solved it using a linear programming (LP) approach (see Dyckhoff [2]). A major advance in CSP solution methods was from Gilmore and Gomory [5,6]. They employed a delayed pattern generation technique and LP approach to effectively deal with cutting pattern

generation and solution of the CSP. Subsequently, there has been a great deal of research in this area. Various formulations of the CSP considering different criteria and solution techniques have been developed.

Most of the techniques used in solving the one-dimensional CSP or trim problem have applied LP-based algorithms (see for example Daniels and Ghandforoush [1], Dyckhoff [2], Wascher [18], Gilmore and Gomory [5] and [6]). This may be due to the NP-hard complexity of the one-dimensional CSP when variations in final roll size become substantially large. For instance, as pointed out in Dyckhoff [2], with a standard raw roll of 200″ and demands of 40 different final rolls ranging from 20″ to 80″, the number of cutting patterns or variables in the mathematical formulation can easily exceed millions. Although millions of cutting patterns seem unrealistically large, cutting patterns in the range of thousands can be quite realistic in many industrial applications. Such large problem sizes cannot be solved by existing integer programming algorithms when optimal solutions are desired within reasonable computation time. This may explain why the LP approach has remained a widely accepted technique.

In the LP approach, integral solutions are obtained by rounding the values of the decision variables in the optimal solution to the relaxed LP. However, this rounding may lead to suboptimal and even infeasible solutions. When rounding leads to infeasible solutions, additional manipulation of the final solution is required to achieve feasibility. Many different heuristic approaches have been developed to overcome these difficulties. Some generate patterns sequentially to overcome explosion of problem size, and others employ a systematic procedure for rounding of LP solutions. Haessler and Sweeney [8] summarized methods such as sequential heuristic procedures (SHP) and Hybrid solution procedures. The primary advantage of the SHP approach is its ability to control factors other than trim loss and eliminate rounding problems by working with only integer values. The main disadvantages are that it may lead to suboptimal solutions with higher trim loss, and its success depends on intuitive and intelligent choices in early pattern generation. Another approach is the Hybrid Solution Procedure which is the integration of SHP and LP. This method is somewhat superior to SHP in that it can provide either pure SHP or LP solutions. Additionally, it can generate solutions which are partially SHP and partially LP and therefore likely to be more desirable than either one alone.

Goulimis [7] proposed a new three-phase algorithm in which the first phase is pattern generation by either dynamic programming or tree search methods. The second phase is LP-relaxation, and the third phase provides a method of combined cutting planes and branch-and-bound technique for identification of optimal, integral solutions. This heuristic avoids the problems associated with LP-rounding but has the disadvantage of being restricted to solution of problems of relatively small size due to the IP algorithm in the final phase.

Stadtler [16] developed a first-fit-decreasing (FFD) heuristic supplemented by a one-pass branching up procedure to identify integral solutions through LP rounding.

It is important to point out that many of the heuristics developed to date still cannot avoid the need for LP-rounding to achieve integral solutions. It is our opinion that elimination of the LP-rounding process is desirable. Furthermore, if the model includes Boolean, fixed-charge decision variables reflecting the selection or omission of a given cutting pattern (see the model presented later herein), relaxed LP solutions will typically not reflect the effects of the fixed-charge on the optimal solution. The desire to overcome the drawbacks of LP-rounding, the inclusion of fixed-charge criteria in our model, and the desire to handle large problem sizes has motivated us to explore the application of a simulated annealing based heuristic to find a near-optimal solution for our ILP model of a one-dimensional CSP.

## 2. Model formulation

The following formulation is adapted from Daniels and Ghandforoush [1] with the inclusion of stocking limits on different final roll sizes and the inclusion of fixed-charge (set-up cost) criteria. A particular cut can result in different final roll sizes or final rolls with the same size.

The formulation assumes that customer orders are not handled individually. That is, the model is based

on a list which summarizes all customer requirements for a specific production period (see Daniels and Ghandforoush [1]).

### 2.1. Define

$X_j$: Number of lots to be made with set-up number $j$.
$A_{ij}$: $i^{\text{th}}$ size of the final roll in the $j^{\text{th}}$ set-up.
$C_j$: Trim loss cost per cut associated with the $j^{\text{th}}$ set-up.
$B_i$: Total order quantity for $i^{\text{th}}$ size of final roll.
$S_j$: Machine time/cut.
$Q_i$: Stocking limit of the $i^{\text{th}}$ size of final roll.
$Y_j$: Indicator variable (1 if set-up $j$ used; otherwise, 0).
$F_j$: Cost for $j^{\text{th}}$ set-up.
$T$: Maximum difference in total production time.
$D$: Difference of total production time between machines.
$M$: A large, integral constant.

Given this, the model is expressed:

$$\min \sum_j C_j X_j + \sum_j F_j Y_j$$

subject to

$$\sum_j A_{ij} X_j \geq B_i \quad \forall i, \tag{1}$$

$$\sum_j A_{ij} X_j \leq B_i + Q_i \quad \forall i, \tag{2}$$

$$|D| \leq T \quad \forall \text{ pairs of machines}, \tag{3}$$

$$X_j - MY_j \leq 0 \quad \forall j \,(\text{where } M \gg 1), \tag{4}$$

$$X_j \geq 0 \text{ and integral } \forall_j \tag{5}$$

$$Y_j \in \{0,1\} \quad \forall j, \tag{6}$$

where

$$D = S_i \sum_j X_j - S_{j'} \sum_{j'} X_{j'},$$

($j$ and $j'$ denote two distinct machines).

In the above formulation, the objective is to minimize the total cost which is the sum of the total trim loss costs and the total set-up costs. The set-up costs are machine dependent. For example, machines with different ages will have different set-up costs. Efficiency of machines and required set-up times also contribute to the difference in set-up costs. The trim loss costs can be obtained from a loss function which varies for different industrial applications. Usually the loss function is based on the per roll inch price, realizable price, or recyclability of the material. Recycling material typically results in a lower grade of such material and a discounted price must therefore be reflected in the loss function.

The first set of constraints (1) ensure the fulfilment of customer orders; whereas, the second set (2) imposes the stocking limit for each final roll size. If the planned production period is based on the actual customer orders rather than forecasted demands, the first set of constraints is difficult to formulate. The second constraints may require adjustment when the solution to the model results in an unsatisfactorily high cost. Relaxation of the right-hand side can often yield lower-cost solutions. However, this results in an undesirable increase in the final roll stock. Typically, a balance should be achieved between the two.

The third set of constraints (3) ensures a balanced load between machines. These constraints can ensure utilization of machines, indirectly eliminate the possibility of undesirable increases in lead time, and avoid the use of only a limited number of material sizes.

The fourth set of constraints (4) impose a fixed-charge indicator, $Y_j$, associated with the cutting-pattern variable, $X_j$. Specifically,

$$Y_j = \begin{cases} 0 & \text{if } X_j = 0 \\ 1 & \text{otherwise.} \end{cases}$$

The above formulation can accommodate many variations associated with different industrial applications. The following are a few examples:

(1) If management does not want to have any stock of final rolls, constraints (2) can be eliminated, and the right-hand side of (1) becomes a strict equality. One possible consequence of this is a reduction in solution space and a larger, optimal objective value.
(2) When dealing with high-value material, constraints (3) can be relaxed with the assumption of identical machines simultaneously running identical set-ups. This procedure will shorten the lead time. Also, if the value of the material is sufficiently high, the trim loss costs will become

dominant in the objective function thereby making the set-up costs negligible. In this case, a new formulation without the set-up criteria may be more applicable.

In this paper, we limit our study to the application of the simulated annealing heuristic to the original formulation.

## 3. The cutting stock algorithm

The complete algorithm for the cutting stock model presented herein consists of three phases: (i) The generation of cutting patterns, (ii) the generation of an initial feasible solution to the ILP, and (iii) the application of simulated annealing to the solution of the ILP model. A detailed description of these steps follows.

*Step 1: Pattern Generation*

We are using an enumerative search technique to identify all possible patterns (the $X_j$s in the model). Although many previous works have suggested that cutting patterns with high loss can be ignored, this may lead to suboptimization; moreover, the relatively few cutting patterns with low trim loss may not satisfy all constraints. Therefore, given the existence of an efficient heuristic capable of facilitating large problem instances, the omission of high trim loss cutting patterns is unnecessary.

*Step 2: Generate Initial Feasible Solution (IFS)*

The purpose of generating an IFS is to provide a starting solution in the simulated annealing process of Step 3. Although an IFS is not mandatory, experiments using the algorithms described in the following text clearly demonstrated that without an IFS the convergence of the simulated annealing algorithm is not guaranteed, and the result is generally poor as compared to one employing an IFS.

The IFS for our algorithm was provided by the first incumbent solution of a branch-and-bound procedure. For this step, the set-up variables, $Y_j$, and their associated constraints in (4) of the mathematical model – neither of which affects the feasibility of the IFS – were dropped from the model to expedite identification of the first incumbent solution. (Note that branch-and-bound procedures typically provide rapid convergence to the first incumbent solution. A result supported by our experimentation.)

*Step 3: Simulated Annealing*

The heuristic employed in this research is a slight variation of the basic simulated annealing algorithm presented in the previous section. Initially, we experimented with a procedure which has been reported in the literature (see Isken and Hancock [11]) to provide good results for certain ILPs (hereafter, The Existing Method). Briefly, that procedure increments a randomly selected decision variable by one, decrements another (distinct) randomly selected decision variable by one, assigns an appropriate penalty cost for violation of feasibility, evaluates the energy function, and accepts or rejects the incumbent solution as outlined in Step 3 of the general simulated annealing procedure. However, we found this procedure resulted in convergence (or freezing) at a suboptimal state. Based on experimentation with the two methods mentioned, we conjectured that the unit increase/decrease of a pair of decision variables was insufficient to ensure movement from a local optimum. Since upper and lower bounds on all decision variables for the ILP formulation are known, we decided to try setting a pair of randomly selected decision variables to a value uniformly distributed between their respective lower and upper bounds. The constraints were then evaluated for the trial solution. If the trial solution was feasible, the energy function was evaluated; however, rather than rejecting the trial solution if the constraints were violated, we determined the range of feasible values for one of the decision variables (the other was reset to its initial value) and set its value randomly from a uniform distribution over the established feasible range. The energy function was then evaluated with this trial solution. This procedure performed well and was accepted as the final form of the algorithm. (Note: A computational comparison of the two methods is provided in Section 5.)

The steps of our final implementation of the simulated annealing heuristic are outlined below. The algorithm was implemented in the C programming language using Turbo C Version 2.0 on an IBM-compatible 486DX2 66 MHz personal computer. In the algorithm, $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ denotes the vector of decision variables, LB and UB denote the lower and upper bounds on all decision

variables, respectively, and U($a$, $b$) denotes a variate from a discrete, uniform distribution over the range $[a, b]$.

### 3.1. Simulated annealing ILP algorithm (SAILP)

*Step 1.* Let $n$ denote the number of non fixed-charge decision variables. Select an initial temperature, $t$ = TINIT, and an initial feasible solution (as previously described), $x_0$. Let $f_0 = f(x_0)$ denote the corresponding objective value. Set $i = 0$, NSIZE = $n$, ITER = NSIZE(SIZEFACTOR), $n$ = ITER, and go to Step 2.

*Step 2.* Set $i = i + 1$. Randomly select two decision variables, $x_{ij}$ and $x_{ik}$, and set $x_{ij} \sim$ U(LB, UB) and $x_{ik} \sim$ U(LB, UB). If the resulting decision vector, $x_i$, is feasible, evaluate $f_i = f(x_i)$ and go to Step 3. Otherwise, set $x_{ik} = x_{i-1,k}$ and evaluate the constraints for determination of a feasible lower bound, say XLB, and a feasible upper bound, say XUB, on decision variable $x_{ij}$. Set $x_{ij} \sim$ U(XLB, XUB), evaluate $f_i = f(x_i)$, and go to Step 3.

*Step 3.* If $f_i < f_{i-1}$, then set FAILCOUNT = 0 and go to Step 5. Otherwise, accept $f_i$ as the new solution with probability

$$e^{|f_i - f_{i-1}|/t}$$

and go to Step 4.

*Step 4.* If $f_i$ was rejected as the new solution in Step 3, then set REJECTED = REJECTED + 1, $f_i = f_{i-1}$, and $x_i = x_{i-1}$. If $f_i$ was accepted in Step 3, then set ACCEPTED = ACCEPTED + 1. Go to Step 5.

*Step 5.* If satisfied with the current objective value, $f_i$, stop. Otherwise, set $n = n - 1$. If $n > 0$, then go to Step 2. Otherwise, set $t = t \cdot$ TFAC. If $t <$ TSTOP, then stop. Else if

$$\left(\frac{\text{ACCEPTED}}{\text{ACCEPTED} + \text{REJECTED}}\right) \leq \text{MINACCEPT},$$

set FAILCOUNT = FAILCOUNT + 1. If FAILCOUNT = FREEZEMAX, then stop; otherwise, set $n$ = ITER, REJECTED = 0, and go to Step 2.

It should be noted that the constraints in the original ILP formulation associated with the fixed-charge variables need not be directly evaluated. That is, each fixed-charge variable is set prior to evaluation of the objective function. It is clear that at most two fixed charge variables will be evaluated per iteration of the algorithm – resulting in a significant decrease in computational burden. The execution time of the algorithm is further decreased by partial evaluation of the objective function at each iteration. That is, we need only consider the effect (with respect to the objective function) of those decision variables which were altered in the trial solution.

## 4. Parameter selection

In this section, the experiments conducted to determine the "ideal" parameters for the annealing algorithm are described. The procedures followed were first presented by Johnson et al. [12] and have been applied by Hart and Chen [10]. Due to the number of parameters and their inherent interactions, a complete enumeration and analysis is not possible; however, the methodology presented herein facilitates a sufficient investigation of the effects of various parameters on the algorithm's performance.

Referring to the SAILP algorithm, the annealing parameters of interest are (i) TINIT – the initial starting temperature of the algorithm, (ii) SIZEFACTOR – the number of trial solutions per temperature level, (iii) TFAC – the rate at which the algorithm's temperature is decreased, (iv) MINACCEPT – threshold corresponding to the minimum number of allowable accepted moves at a given temperature level, and (v) TSTOP – the algorithm's termination temperature. An additional parameter of interest is INITPROB which is a dependent variable representing the probability of accepting a trial solution during the first temperature level. In SAILP, INITPROB is expressed as the proportion of nonimproving moves accepted by the algorithm during the first temperature level and is clearly dependent upon TINIT and SIZEFACTOR.

The test problem employed in the design of the algorithm's parameter set and in the computational comparison of SAILP with the existing SA-based technique was a cutting stock problem consisting of four machines and customer orders of four final roll sizes. The data for this problem can be found in the Appendix. The test problem was modeled using the ILP formulation presented herein, and its optimal

solution was obtained using the LINDO optimization package. We chose a "small" cutting stock problem to ensure that the optimal solution would be available to evaluate the performance (with respect to objective value) of the SAILP algorithm. Since our primary purpose in this research was to compare the efficiency of SAILP with the existing SA-based ILP methodology, we felt that size of the problem solved was not a primary issue.

Where applicable, a statistical analysis of the experimental results has been performed. Analysis of Variance (ANOVA) is the most common technique for determining if a given factor will significantly affect the outcome of an experiment, interactions between factors, and optimal conditions. A basic assumption of ANOVA is that the variances of all the treatment combinations are equal; however, according to the results of our experiments, this assumption is not satisfied. Therefore, ANOVA cannot be employed in the statistical analysis. Instead, the $t$-test has been employed to test the null hypothesis that the output of two treatment combinations are equal (i.e., $\mu_1 - \mu_2 = 0$) since it can be employed when the variances of the output of the treatment combinations are equal and when they are unequal (For unequal variances, the test is referred to as the Smith–Satterthwaite test). Specifically, for equal variances the statistic is defined:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}} \times \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}}$$

with $n_1 + n_2 - 2$ degrees of freedom.

For unequal variances:

$$t' = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \text{ with}$$

$$v = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1 + 1} + \frac{(s_2^2/n_2)^2}{n_2 + 1}}$$

degrees of freedom.

To determine which test will be used in the analysis, an $F$-test for determining if the variances of the treatment combinations are equal (i.e., $\sigma_1^2 = \sigma_2^2$) must be performed:

$$F = \frac{\max\{s_1^2, s_2^2\}}{\min\{s_1^2, s_2^2\}}$$

with $n_1 - 1$ and $n_2 - 1$ degrees of freedom.

(For all $F$-tests, the alternative hypothesis is that the variance of the treatment combination associated with the numerator in the expression above is greater than that associated with the denominator, and all tests were performed first at the 0.01 level of significance: $F_{0.01}(39,39) = 2.12$. If the null hypothesis could not be rejected at this level of significance, then the test was performed at the 0.05 level of significance: $F_{0.05}(39,39) = 1.70$. An examination of the data presented in the following discussions indicates that the null hypothesis was rejected at the 0.01 level of significance for many of the treatment combinations supporting the necessity to rely on the $t$-tests rather than ANOVA for data analysis.)

The first experiment conducted established a relationship between TINIT and the acceptance probability, INITPROB. For this experiment, the parameters SIZEFACTOR, TFAC, MINACCEPT, and FREEZEMAX were set at 8, 0.95, 0.02, and 5, respectively. Forty replications for various values of TINIT were conducted, and the values of INITPROB were calculated and averaged over the 40 replications. The results are depicted in Table 1.

The second experiment conducted examined the effect of INITPROB on the efficiency (with respect to objective value and execution time) of the algorithm. For this experiment, 40 replications of the algorithm were conducted using the TINIT values of Table 1 and the aforementioned values for SIZEFACTOR, TFAC, MINACCEPT, and FREEZEMAX with TSTOP = 0.001. The results of these experiments appear in Table 2. The $t$-test row con-

Table 1
Relationship between initial temperature (TINIT) and the acceptance probability (INITPROB)

| TINIT | 7500 | 2900 | 1500 | 700 |
|---|---|---|---|---|
| INITPROB | 0.9 | 0.8 | 0.7 | 0.6 |

Table 2
Effects of starting temperature (TINIT) and the acceptance probability (INITPROB) using SAILP

| | TINIT/INITPROB | | | |
|---|---|---|---|---|
| | 7500/0.9 | 2900/0.8 | 1500/0.7 | 700/0.6 |
| CPU time (sec) | 266.89 | 254.75 | 247.32 | 240.79 |
| Mean objective value | 24176.25 | 23472.88 | 25723.13 | 32032.25 |
| Standard deviation | 6584.10 | 4648.89 | 5541.77 | 1279.03 |
| $t$-test | $F = 2.01$ | – | $F = 1.42$ | $F = 13.21$ |
| | $t' = 0.55$ | – | $t = 1.96$ | $t' = 11.23$ |
| | $\alpha = 0.05$ | – | $\alpha = 0.05$ | $\alpha = 0.005$ |
| | $\nu = 72$ | – | – | $\nu = 44$ |

tains information related to the test of the null hypothesis that the mean objective value of the (TINIT, INITPROB) = (2900, 0.8) pair was equal to that of the other (TINIT, INITPROB) pairs. The value of the test statistic is reported along with the level of significance, $\alpha$, employed in the test. When the value, $\nu$, is reported, the $F$-test indicated a significant difference in the variances and hence the Smith–Satterthwaite test was applied. Otherwise, the $t$-test was applied.

Based upon the results of the statistical analysis, we can conclude that the mean objective value obtained using the (TINIT, INITPROB) = (2900, 0.8) pair is significantly less than that obtained with the (1500, 0.7) pair at a 0.05 level of significance and that obtained with the (700, 0.6) pair at a 0.0005 level of significance. Although there is no statistically significant difference at the 0.05 level of significance between the (2900, 0.8) and (7500, 0.9) pairs, the lower variance and execution times of the (2900, 0.8) pair favor it for inclusion in the standard parameter set.

The effects of TFAC and SIZEFACTOR were analyzed by conducting 40 replications of the algorithm using the (TFAC, SIZEFACTOR) pairs depicted in Table 3 (all other parameters were fixed at their aforementioned values).

The values of TFAC were chosen such that each increase in value represents the square root of the previous value which should yield a doubling of the number of temperature levels at which the objective function is evaluated. Similarly, each increase in SIZEFACTOR results in a doubling of the number of trial solutions per temperature level. As noted by Johnson et al. [12], fixing one parameter and increasing the other to its next value should yield an approximate doubling of the algorithm's execution time. The results of Table 3 support this observation.

From the table, increasing SIZEFACTOR to its next value while fixing TFAC yields an average 9.9% decrease in the objective value; whereas, increasing TFAC to its next value while fixing SIZE-

Table 3
Effects of TFAC and SIZEFACTOR on average CPU time, objective value, and standard deviation using SALIP

| | CPU time (secs)<br>Objective value<br>Standard deviation | | |
|---|---|---|---|
| | TFAC | | |
| SIZEFACTOR | 0.9025 | 0.9500 | 0.9747 |
| 1 | 16.09 | 32.09 | 65.64 |
| | 40493.50 | 34471.38 | 29955.25 |
| | 5814.46 | 3883.97 | 5147.09 |
| 2 | 32.18 | 64.09 | 125.86 |
| | 33019.00 | 28856.00 | 24659.00 |
| | 4616.23 | 4554.44 | 4870.75 |
| 4 | 65.71 | 125.73 | 260.33 |
| | 27949.25 | 25401.25 | 22425.00 |
| | 4887.03 | 5049.53 | 4162.22 |
| 8 | 128.55 | 254.75 | 511.29 |
| | 24595.13 | 23472.88 | 22099.25 |
| | 4568.84 | 4648.89 | 4262.22 |
| 16 | 257.15 | 513.23 | 1025.55 |
| | 22346.88 | 20248.13 | 20323.25 |
| | 3433.53 | 778.06 | 773.73 |
| 32 | 513.15 | 1016.88 | 2019.67 |
| | 20686.38 | 20100.25 | 20274.63 |
| | 1122.42 | 458.46 | 124.10 |

Table 4
Statistical analysis of improvement in objective value for various (TFAC, SIZEFACTOR) treatments

| | | 32 | | | 16 | | |
|---|---|---|---|---|---|---|---|
| | | 0.9747 | 0.9500 | 0.9025 | 0.9747 | 0.9500 | 0.9025 |
| 32 | 0.9747 | | $t' = 2.23$<br>$\alpha = 0.05$<br>$\nu = 45$ | $t' = 2.31$<br>$\alpha = 0.05$<br>$\nu = 40$ | $t' = 0.39$<br>$\alpha = 0.05$<br>$\nu = 42$ | $t'0.21$<br>$\alpha = 0.05$<br>$\nu = 42$ | $t' = 3.81$<br>$\alpha = 0.0005$<br>$\nu = 40$ |
| | 0.9500 | – | – | $t' = 3.06$<br>$\alpha = 0.005$<br>$\nu = 53$ | $t' = 1.56$<br>$\alpha = 0.05$<br>$\nu = 65$ | $t' = 1.04$<br>$\alpha = 0.05$<br>$\nu = 65$ | $t' = 4.10$<br>$\alpha = 0.0005$<br>$\nu = 41$ |
| | 0.9025 | – | – | – | $t' = 2.11$<br>$\alpha = 0.05$<br>$\nu = 71$ | $t' = 2.03$<br>$\alpha = 0.05$<br>$\nu = 71$ | $t' = 2.91$<br>$\alpha = 0.005$<br>$\nu = 48$ |
| 16 | 0.9747 | – | – | – | – | $t = 0.43$<br>$\alpha = 0.05$<br>– | $t' = 3.64$<br>$\alpha = 0.0005$<br>$\nu = 44$ |
| | 0.9500 | – | – | – | – | – | $t' = 3.77$<br>$\alpha = 0.0005$<br>$\nu = 44$ |

FACTOR yields an average 8.1% decrease in objective value. These results indicate that the efficiency (with respect to objective value) of the algorithm is improved by increasing the algorithm's run time by either increasing the number of trials per temperature level or by increasing the number of temperature levels. This conclusion was verified by performing $t$-tests between TFAC columns and SIZEFACTOR row pairs for SIZEFACTOR values of 16 and 32 in Table 3. The results are tabulated in Table 4. From the table, 11 of the 15 treatment combinations (73%) indicate a statistically significant difference between average objective values.

Given these results, it was decided that the standard settings for TFAC and SIZEFACTOR should represent a reasonable tradeoff of efficiency for improvement in execution time. From Table 3, the best objective value was obtained when TFAC = 0.95 and SIZEFACTOR = 32; however, TFAC = 0.95 and SIZEFACTOR = 16 provides an average objective value only 0.73% higher which, based upon the $t$-test in Table 4, is statistically insignificant at the 0.05 level of significance. The $t$-tests of Table 4 also indicate that SIZEFACTOR values greater than 32 need not be considered since there is no statistically

Table 5
Statistical analysis of average objective value for (TFAC, SIZEFACTOR) = (0.95, 16) pair versus average objective value obtained for treatments with lower CPU times

| | TFAC | | |
|---|---|---|---|
| SIZEFACTOR | 0.9025 | 0.9500 | 0.9747 |
| 1 | $t' = 21.83$<br>$\alpha = 0.0005$<br>$\nu = 41$ | $t' = 22.71$<br>$\alpha = 0.0005$<br>$\nu = 43$ | $t' = 11.79$<br>$\alpha = 0.0005$<br>$\nu = 41$ |
| 2 | $t' = 17.25$<br>$\alpha = 0.0005$<br>$\nu = 42$ | $t' = 11.75$<br>$\alpha = 0.0005$<br>$\nu = 42$ | $t' = 5.66$<br>$\alpha = 0.0005$<br>$\nu = 42$ |
| 4 | $t' = 9.84$<br>$\alpha = 0.0005$<br>$\nu = 42$ | $t' = 6.38$<br>$\alpha = 0.0005$<br>$\nu = 41$ | $t' = 3.25$<br>$\alpha = 0.005$<br>$\nu = 42$ |
| 8 | $t' = 5.93$<br>$\alpha = 0.0005$<br>$\nu = 42$ | $t' = 4.33$<br>$\alpha = 0.0005$<br>$\nu = 42$ | $t' = 2.70$<br>$\alpha = 0.005$<br>$\nu = 42$ |
| 16 | $t' = 3.77$<br>$\alpha = 0.0005$<br>$\nu = 44$ | – | – |
| 32 | $t' = 2.3$<br>$\alpha = 0.05$<br>$\nu = 48$ | – | – |

Table 6
Standard parameters chosen for the SAILP algorithm based upon statistical analysis

| TINIT | TFAC | SIZEFACTOR | MINACCEPT | TSTOP |
|---|---|---|---|---|
| 2900 | 0.9500 | 16 | 0.02 | 0.001 |

significant difference between the average objective values obtained for SIZEFACTOR = 16 and SIZEFACTOR = 32 treatments with a given TFAC. Therefore, since the average execution time of the TFAC = 0.95 and SIZEFACTOR = 16 pair is half that of the TFAC = 0.95 and SIZEFACTOR = 32 pair with no statistically significant difference in average objective value, the TFAC = 0.95 and SIZEFACTOR = 16 pair was selected as standard settings. Table 5 illustrates that the average objective value obtained with (TFAC, SIZEFACTOR) = (0.95, 16) pair dominates the average objective value obtained with all other pairs with lower execution times. Hence, a reasonable tradeoff is accomplished with the (0.95, 16) pair.

The final experiment required determination of the stopping criteria, TSTOP and MINACCEPT. Referring to SAILP, the algorithm is terminated when either (i) the temperature falls below TSTOP or (ii) the number of accepted moves falls below MINACCEPT for FREEZEMAX consecutive temperature levels. The effect of TSTOP was analyzed by performing 40 replications for various values of TSTOP on the range [0.000001, 0.1]. The results indicated that little or no improvement in objective value was obtained by decreasing TSTOP below 0.0001. Although increasing TSTOP above 0.0001 yielded shorter execution times, the average objective values were deemed unacceptably high with respect to those obtained with TSTOP = 0.0001. In all experiments, the value of MINACCEPT was shown to have no effect on the algorithm (This observation held for various values of FREEZEMAX). That is, the algorithm always terminated when the temperature fell below TSTOP.

Based on the aforementioned experiments, the parameter values shown in Table 6 were selected as the standard settings.

## 5. Computational comparison

In this section, a computational comparison between SAILP and the existing SA-based technique for ILP problems is presented. The analysis was conducted on the same problem employed in the selection of SAILP's standard parameter settings.

To facilitate comparison between SAILP and the Existing Method required determination of an "optimal" TINIT value for the Existing Method and exploration of its performance for various (TFAC, SIZEFACTOR) pairs using this TINIT parameter; hence, the experiments reported for SAILP in Tables 2 and 3 were repeated for the Existing Method. The results of these experiments appear in Tables 7 and 8, respectively.

Based upon the results of the statistical analysis reported in Table 7, we can conclude that the mean objective value obtained using the (TINIT, INITPROB) = (2050, 0.8) pair is significantly less than that obtained with the (5000, 0.9) and (1050, 0.7) pairs at a 0.05 level of significance and that obtained with the (200, 0.6) pair at a 0.0005 level of signifi-

Table 7
Effects of starting temperature (TINIT) and the acceptance probability (INITPROB) using the existing method

| | TINIT/INITPROB | | | |
|---|---|---|---|---|
| | 5000/0.9 | 2050/0.8 | 1050/0.7 | 200/0.6 |
| CPU time (sec) | 254.59 | 246.39 | 241.56 | 212.12 |
| Mean objective value | 39070.38 | 37267.00 | 38795.25 | 42120.88 |
| Standard deviation | 4656.64 | 3686.63 | 4190.94 | 6489.18 |
| $t$-test | $F = 1.60$ | – | $F = 1.29$ | $F = 3.10$ |
| | $t = 1.92$ | – | $t = 1.73$ | $t' = 4.11$ |
| | $\alpha = 0.05$ | – | $\alpha = 0.05$ | $\alpha = 0.0005$ |
| | – | – | – | $\nu = 44$ |

Table 8
Effects of TFAC and SIZEFACTOR on average CPU time, objective value, and standard deviation using the existing method

| | CPU time (secs)<br>Objective value<br>Standard deviation | | |
|---|---|---|---|
| | TFAC | | |
| SIZEFACTOR | 0.9025 | 0.9500 | 0.9747 |
| 1 | 15.97 | 30.14 | 62.16 |
| | 50113.00 | 51981.00 | 47996.00 |
| | 3759.37 | 3704.88 | 5298.62 |
| 2 | 30.66 | 62.19 | 124.68 |
| | 50688.88 | 48015.75 | 43827.25 |
| | 4837.63 | 4868.96 | 4664.78 |
| 4 | 61.17 | 123.48 | 248.59 |
| | 48251.00 | 40922.00 | 36972.00 |
| | 4461.49 | 5373.62 | 3468.56 |
| 8 | 125.15 | 246.39 | 500.05 |
| | 42655.00 | 37267.00 | 34150.13 |
| | 4920.58 | 3686.63 | 3047.61 |
| 16 | 250.79 | 496.02 | 990.65 |
| | 38421.38 | 34008.63 | 28706.38 |
| | 4097.65 | 2274.56 | 5732.19 |
| 32 | 502.22 | 985.87 | 1945.72 |
| | 33891.25 | 28503.75 | 21866.75 |
| | 2661.34 | 5410.85 | 2083.15 |

cance; therefore, TINIT = 2050 was employed in the generation of the (TFAC, SIZEFACTOR) analysis of Table 8.

Given the "optimal" TINIT values for the SAILP algorithm (identified in the analysis of Table 2) and the Existing Method (identified in the analysis of Table 7), the first performance comparison between the two algorithms involved a statistical analysis of the average objective values for all (TFAC, SIZEFACTOR) combinations in Tables 3 and 8. (i.e., the performance of the two algorithms with respect to average objective value was analyzed under identical conditions.) These results are reported in Table 9.

From the table, the performance of SAILP dominates that of the Existing Method at the 0.0005 level of significance for all (TFAC,SIZEFACTOR) combinations. Finally, Table 10 compares the average objective values obtained for all TFAC values when

Table 9
Statistical analysis of average objective values obtained under identical (TFAC, SIZEFACTOR) combinations for SAILP and the existing method

| | TFAC | | |
|---|---|---|---|
| SIZEFACTOR | 0.9025 | 0.9500 | 0.9747 |
| 1 | $t' = 8.84$<br>$\alpha = 0.0005$<br>$\nu = 69$ | $t = 20.63$<br>$\alpha = 0.0005$<br>– | $t = 15.45$<br>$\alpha = 0.0005$<br>– |
| 2 | $t = 16.71$<br>$\alpha = 0.0005$<br>– | $t = 19.46$<br>$\alpha = 0.0005$<br>– | $t = 17.98$<br>$\alpha = 0.0005$<br>– |
| 4 | $t = 19.40$<br>$\alpha = 0.0005$<br>– | $t = 13.31$<br>$\alpha = 0.0005$<br>– | $t = 16.98$<br>$\alpha = 0.0005$<br>– |
| 8 | $t = 17.01$<br>$\alpha = 0.0005$<br>– | $t = 14.01$<br>$\alpha = 0.0005$<br>– | $t = 14.55$<br>$\alpha = 0.0005$<br>– |
| 16 | $t = 19.01$<br>$\alpha = 0.0005$<br>– | $t' = 36.20$<br>$\alpha = 0.0005$<br>$\nu = 49$ | $t' = 9.17$<br>$\alpha = 0.0005$<br>$\nu = 41$ |
| 32 | $t' = 28.91$<br>$\alpha = 0.0005$<br>$\nu = 54$ | $t' = 9.78$<br>$\alpha = 0.0005$<br>$\nu = 40$ | $t' = 4.82$<br>$\alpha = 0.0005$<br>$\nu = 40$ |

the execution time of SAILP is approximately $1/8^{th}$ that of the Existing Method.

From the table, the performance of SAILP (with respect to average objective value) dominates that of the Existing Method even when the execution time of SAILP is significantly less than that of the Existing Method.

Table 10
Statistical analysis of average object values obtained when the execution time of SAILP is approximately $1/8^{th}$ that of the existing method

| SIZEFACTOR | TFAC | | |
|---|---|---|---|
| SAILP/Existing method | 0.9025 | 0.9500 | 0.9747 |
| 1/8 | $t = 1.8$<br>$\alpha = 0.05$<br>– | $t = 3.53$<br>$\alpha = 0.0005$<br>– | $t' = 4.44$<br>$\alpha = 0.0005$<br>$\nu = 65$ |
| 2/16 | $t = 5.54$<br>$\alpha = 0.0005$<br>– | $t' = 6.40$<br>$\alpha = 0.0005$<br>$\nu = 59$ | $t = 3.40$<br>$\alpha = 0.005$<br>– |

## 6. Conclusions

This paper has presented a new SA-based solution algorithm for ILP problems. Extensive analysis of the effects of various parameters on the accuracy and efficiency of solutions generated by the algorithm has been provided. A new ILP model of the cutting stock problem has been developed and employed to illustrate the superior performance of SAILP over an existing SA-based methodology. Specifically, a direct computational comparison and statistical analysis between SAILP and the existing simulated annealing method indicates that SAILP can provide superior solutions (with respect to objective value) in significantly less time.

Further research should be conducted to analyze the performance of SAILP on larger cutting stock problems and to determine the applicability of SAILP to other ILP problems with non binary decision variables.

## Acknowledgements

The authors are grateful to three anonymous referees for many helpful suggestions for improvement of an earlier version of this manuscript.

## Appendix A. Test problem data

Table A1
Machine data

| | Machine | | | |
|---|---|---|---|---|
| | A | B | C | D |
| Machine roll size (inches) | 100 | 80 | 70 | 140 |
| Cutting time (min/cut), $S_j$ | 3 | 4 | 6 | 2 |
| Set-up cost per pattern change (¢/cut), $F_j$ | 1500 | 1200 | 1000 | 2200 |

Table A2
Customer orders

| | | | | |
|---|---|---|---|---|
| Final roll sizes (inches), $i = 1,2,3,4$ | 45 | 36 | 31 | 14 |
| Number of rolls required, $B_i$ | 800 | 300 | 70 | 200 |
| Stocking limit on final rolls, $Q_i$ | 200 | 100 | 50 | 90 |

Table A3
Trim loss costs

| | Loss of material per roll | Realizable price per roll | Net loss per roll |
|---|---|---|---|
| For loss $t \leq 7''$ | $30t$ | 5 | $30t$-5 |
| For loss $t \leq 8''$ | $30t$ | $5t$ | $25t$ |

Table A5
Optimal solution found by integer programming optimum objective value, $F = 19425$

| | |
|---|---|
| $X_6 = 35$ | $Y_6 = 1$ |
| $X_{27} = 80$ | $Y_{27} = 1$ |
| $X_{28} = 280$ | $X_{28} = 1$ |

Table A4
Cutting patterns for test problem

| $j^{th}$ set-up | Machine | Raw width (in) | No. of final rolls | | | | Trim loss (in) | Cost |
|---|---|---|---|---|---|---|---|---|
| | | | $i=1$ 45″ | $i=2$ 36″ | $i=3$ 31″ | $i=4$ 14″ | | |
| 1 | A | 100 | 2 | 0 | 0 | 0 | 10 | 250 |
| 2 | A | 100 | 1 | 1 | 0 | 1 | 5 | 145 |
| 3 | A | 100 | 1 | 0 | 1 | 1 | 10 | 250 |
| 4 | A | 100 | 1 | 0 | 0 | 3 | 13 | 325 |
| 5 | A | 100 | 0 | 2 | 0 | 2 | 0 | 0 |
| 6 | A | 100 | 0 | 1 | 2 | 0 | 2 | 55 |
| 7 | A | 100 | 0 | 1 | 1 | 2 | 5 | 145 |
| 8 | A | 100 | 0 | 1 | 0 | 4 | 8 | 200 |
| 9 | A | 100 | 0 | 0 | 3 | 0 | 7 | 205 |
| 10 | A | 100 | 0 | 0 | 2 | 2 | 10 | 250 |
| 11 | A | 100 | 0 | 0 | 1 | 4 | 13 | 325 |
| 12 | A | 100 | 0 | 0 | 0 | 7 | 2 | 55 |
| 13 | B | 80 | 1 | 0 | 1 | 0 | 4 | 115 |
| 14 | B | 80 | 1 | 0 | 0 | 2 | 7 | 205 |
| 15 | B | 80 | 0 | 2 | 0 | 0 | 8 | 200 |
| 16 | B | 80 | 0 | 1 | 1 | 0 | 13 | 325 |
| 17 | B | 80 | 0 | 1 | 0 | 3 | 2 | 55 |
| 18 | B | 80 | 0 | 0 | 2 | 1 | 4 | 115 |
| 19 | B | 80 | 0 | 0 | 1 | 3 | 7 | 205 |
| 20 | B | 80 | 0 | 0 | 0 | 5 | 10 | 250 |
| 21 | C | 70 | 1 | 0 | 0 | 1 | 11 | 275 |
| 22 | C | 70 | 0 | 1 | 0 | 2 | 6 | 175 |
| 23 | C | 70 | 0 | 1 | 1 | 0 | 3 | 85 |
| 24 | C | 70 | 0 | 0 | 2 | 0 | 8 | 200 |
| 25 | C | 70 | 0 | 0 | 1 | 2 | 11 | 275 |
| 26 | C | 70 | 0 | 0 | 0 | 5 | 0 | 0 |
| 27 | D | 140 | 3 | 0 | 0 | 0 | 5 | 145 |
| 28 | D | 140 | 2 | 1 | 0 | 1 | 0 | 0 |
| 29 | D | 140 | 2 | 0 | 1 | 1 | 5 | 145 |
| 30 | D | 140 | 2 | 0 | 0 | 3 | 8 | 200 |
| 31 | D | 140 | 1 | 2 | 0 | 1 | 9 | 225 |
| 32 | D | 140 | 1 | 1 | 1 | 2 | 0 | 0 |
| 33 | D | 140 | 1 | 0 | 3 | 0 | 2 | 55 |
| 34 | D | 140 | 1 | 0 | 2 | 2 | 5 | 145 |
| 35 | D | 140 | 1 | 0 | 1 | 4 | 8 | 200 |
| 36 | D | 140 | 1 | 0 | 0 | 6 | 11 | 275 |
| 37 | D | 140 | 0 | 3 | 1 | 0 | 1 | 25 |
| 38 | D | 140 | 0 | 3 | 0 | 2 | 4 | 115 |
| 39 | D | 140 | 0 | 2 | 2 | 0 | 6 | 175 |
| 40 | D | 140 | 0 | 2 | 1 | 2 | 9 | 225 |
| 41 | D | 140 | 0 | 1 | 3 | 0 | 11 | 275 |
| 42 | D | 140 | 0 | 1 | 2 | 3 | 0 | 0 |
| 43 | D | 140 | 0 | 1 | 1 | 5 | 3 | 85 |
| 44 | D | 140 | 0 | 1 | 0 | 7 | 6 | 175 |
| 45 | D | 140 | 0 | 0 | 4 | 1 | 2 | 55 |
| 46 | D | 140 | 0 | 0 | 3 | 3 | 5 | 145 |
| 47 | D | 140 | 0 | 0 | 2 | 5 | 8 | 200 |
| 48 | D | 140 | 0 | 0 | 1 | 7 | 11 | 275 |
| 49 | D | 140 | 0 | 0 | 0 | 10 | 0 | 0 |

## References


[1] Daniels, J.J., and Ghandforoush, P., "An improved algorithm for the non-guillotine-constrained cutting-stock problem", *Journal of the Operational Research Society (UK)* 41/2 (1990) 141–149.

[2] Dyckhoff, H., "A new linear programming approach to the cutting stock problem", *Operations Research* 9 (1961) 849–859.

[3] Eglese, R.W., "Simulated annealing: A tool for operational research", *European Journal of Operational Research* 46 (1990) 271–281.

[4] Gemmill, D.D., and Sanders, J.L., "Approximate solutions for the cutting stock 'portfolio' problem", *European Journal of Operational Research* 44/2 (1990) 167–174.

[5] Gilmore, P.C., and Gomory, R.E., "A linear programming approach to the cutting stock problem", *Operations Research* 9 (1961) 849–859.

[6] Gilmore, P.C., and Gomory, R.E., "A linear programming approach to the cutting stock problem", *Operations Research* 11 (1963) 863–888.

[7] Goulimis, C., "Optimal solutions for the cutting stock problem", *European Journal of Operations Research* 44 (1990) 197–208.

[8] Haessler, R.W., and Sweeney, P.E., "Cutting stock problems and solution procedures", *European Journal of Operational Research* 54/2 (1991) 141–150.

[9] Hajeck, B., "Cooling schedules for optimal annealing", *Mathematics of Operations Research* 13 (1988) 311–329.

[10] Hart, S.M., and Chen, C., "Simulated annealing and the mapping problem: A computational study", *Computers and Operations Research 21* 4 (1994) 445–461.

[11] Isken, M.W., and Hancock, W.M., "A heuristic approach to nurse scheduling in hospital units with non-stationary, urgent demand, and a fixed staff size", *Journal of the Society for Health Systems* 2/2 (1990) 24–41.

[12] Johnson, D.S., Aragon, C.R., McGeogh, L.A., and Schevon, C., "Optimization by simulated annealing: An experimental evaluation; Part I graph partitioning", *Operations Research* 37/6 (1989) 865–892.

[13] Kirkpatrick, S., Gelatt, Jr., C.D., and Vecchi, M.P., "Optimization by simulated annealing", *Science* 220 (1983) 671–680.

[14] Lundy, M., and Mees, A., "Convergence of an annealing algorithm", *Mathematical Programming* 34 (1986) 111–124.

[15] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., "Equations of state calculations by fast computing machines", *Journal of Chemical Physics* 21 (1953) 1087–1092.

[16] Stadtler, H., "A one-dimensional cutting stock problem in the aluminum industry and its solution", *European Journal of Operational Research* 44/2 (1990) 209–223.

[17] Van Laarhoven, P.J.M., and Aarts, E.H.L, *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1987.

[18] Wascher, G., "An LP-based approach to cutting stock problem with multiple objectives", *European Journal of Operational Research* 44 (1990) 175–184.