Investigation into the use of meta-heuristics in the optimisation of log positioning during sawmill processing

by
Johan de Villiers du Plessis

"Thesis presented in partial fulfilment of the requirements for the degree Master of Science in Forestry (Wood Products Science) at the Faculty of AgriSciences, Stellenbosch University

Supervisor: Mr. Brand Wessels
Faculty of AgriSciences
Department of Forest and Wood Science

December 2010

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the authorship owner thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Johan de Villiers du Plessis

Date: 24 August 2010

Executive summary:

The percentage yield of sawn timber recovered from logs has a large influence on the profitability of a sawmill. The positioning of the log as it is fed into the primary breakdown saw is one of the factors that impacts on the volume recovery percentage. The log's position can be adjusted by changes in rotation, offset and skewness and depending on the ranges and increments used for these variables, the number of possible combinations can become substantial. In a sawmill the time available to assess possible log positions is limited and different strategies can be followed to arrive at an optimal or close-to-optimal positioning solution without an exhaustive evaluation of solutions. Meta-heuristics are sometimes used to arrive at solutions for combinatorial optimisation problems in a limited time. The effectiveness of this approach on the optimisation of timber volume recovery based on log form is evaluated in this study using sawmill simulation data of sixty SA pine logs.

A new meta-heuristic, for convenience named the Tentacle algorithm, was developed specifically for the problem of log positioning optimisation. The results obtained with the Tentacle algorithm was compared with results from three existing meta-heuristics i.e. the Simulated Annealing algorithm, the Population Based Incremental Learning algorithm and the Particle Swarm Optimisation algorithm, in terms of its efficiency and effectiveness in finding good log positioning solutions in a limited time. A fifth method, that of exhaustively searching smaller, high quality areas around the centered and "horns-up" and "horns-down" positions in the search space was compared to that of using the meta-heuristic algorithms. In terms of volume recovery, the Tentacle algorithm performed, on average, the best of the four algorithms evaluated. However, exhaustive searches in the smaller, high quality areas in the search space, outperformed the algorithms.

Opsomming:

Die herwinningspersentasie van gesaagde planke uit saagblokke het ‘n groot invloed op die winsgewendheid van ‘n saagmeul. Die posisionering van die blok in die primêre saag is een van die faktore wat die herwinningspersentasie beïnvloed. Die blok se posisie kan verstel word deur veranderinge in rotasie, oplyning en skeefheid. Afhangend van die veld -en inkrementgrootte kan die hoeveelheid moontlike kombinasies beduidend wees. In ‘n tipiese saagmeul is die beskikbare tyd om moontlike posisies te evalueer beperk en verskeie strategieë kan gevolg word om optimale of naby-optimale kombinasies te bereik sonder om alle moontlike kombinasies te evalueer. Meta-heuristieke word soms gebruik om in ‘n beperkte tyd oplossings te vind vir kombinatoriese optimeringsprobleme. Die doeltreffendheid van hierdie metode by die optimering van herwinningspersentasie gebaseer op blokvorm is in hierdie studie ondersoek. Dit is met behulp van saagmeulsimulasiedata soos van sestig SA dennehoutblokke verkry, uitgevoer.

‘n Nuwe meta-heuristiek, genaamd die Tentakelalgoritme, is tydens hierdie studie spesifiek vir die probleem van blokposisie-optimering ontwikkel. Die resultate verkry met die Tentakelalgoritme is vergelyk met drie bestaande meta-heuristieke, nl. die “Simulated Annealing”-algoritme, die “Population Based Incremental Learning”-algoritme en die “Particle Swarm Optimisation”-algoritme in terme van doeltreffendheid om goeie blokposisies in ‘n beperkte tyd te vind. ‘n Vyfde metode, die gebruik van ‘n volledige ondersoek van verkleinde versamelings, rondom hoë-kwaliteit areas in die soekarea, is vergelyk met die gebruik van die meta-heuristieke. Hierdie hoë-kwaliteit areas word gevind rondom die gesentreerde “horns-up” en “horns-down” posisies. Die Tentakelalgoritme het gemiddelde die beste herwinningsresultate van die vier meta-heuristieke wat ondersoek was gelewer. Die volledige ondersoek van verkleinde versamelings in die hoë kwaliteit areas het egter die meta-heuristieke oortref.

Acknowledgements

Special thanks to Mr. Brand Wessels and the University of Stellenbosch, Department of Forest and Wood Science for their funding with regards to this study. Also thanks for their patience with the repeated revised due dates of this study.

All glory to God.

Table of Contents

1. Introduction

Sawn timber is widely used globally. The efficient utilization of this resource is not only of global importance but also specifically of importance to the sawmilling industry.

Together with steel and cement, industrial roundwood counts as one of the main raw materials in the world (see Table 1). The only difference with steel and cement is that wood is a sustainable resource and will probably become an even more important raw material in the future.

Table 1. Comparison of annual world production of various raw materials (Schultz, 1993)

	Billion tons	Billion m³
Roundwood	2.1	3.5
Industrial roundwood	1.0	1.7
Cement	1.1	1.0
Steel	0.8	0.1
Plastics	0.09	0.08
Aluminium	0.02	0.007

Trees for sawn timber production can be harvested, either from natural or planted plantations. The logs from these trees are brought to the mills where they are ultimately broken down into boards which are referred to as sawn timber. According to data from FAO (2010), sawn timber accounts for roughly 40% of all manufactured wood based products globally in terms of mass.

The volume yield of boards or sawn timber as a percentage of the log volume can be expressed as volume recovery. For example if one cubic meter of boards are sawn from two cubic meters of logs a 50% volume recovery was attained. Production efficiency can also be measured by the value of the boards recovered. For example if boards to the value of R 1000 are recovered from one cubic meter of logs, the value recovery can be expressed as R 1000 per cubic meter.

Two factors, namely the sustainability of the resource and the profitability of the sawmill demand the greatest possible volume and value recovery from every log. With the advancement in technology, sawmills today have the opportunity to optimise the value or volume recovery for every specific log entering the mill. Instead of sawing logs to generally suitable patterns, logs can now be measured with automated three-dimensional log scanning systems and sawn to its specifically optimised sawing pattern. In this instance pattern refers to the arrangement of the different saw blades as well as the spatial positioning of the log in relation to the saws.

When a log is being sawn at the primary breakdown saw there will be one or possibly more log positions that will optimise the volume or value recovery obtained from that log. Positioning of a log includes the rotation and axial positioning of the log in relation to the saws. The position of the log can thus be described in terms of either two or three variables (when skewing is included as a variable) with each variable having an infinite number of options which can be reduced to discrete options. The number of different possible log positions to consider can run into tens of thousands depending on the positioning range and increments used and finding an optimal position can be a time-consuming task. For automated three-dimensional log scanning and positioning systems a very limited time is available to reach an optimal positioning solution and the number of iterations should be minimized.

This process of optimisation by selecting the most suitable combination of discrete variables is known as combinatorial optimisation. It is typical for a combinatorial optimisation problem that the number of combinations makes it impractical to evaluate each combination. For instance, in a problem where there are 3 variables where each variable have 30 different settings, a total of 303 = 27 000 different combinations will be possible. Heuristics, which will be defined later, can then be used to reduce the number of combinations evaluated and attain “good”, but not necessarily optimal, results in much less time.

This study investigates the optimisation of log positioning in front of the primary breakdown saw. A well-known heuristic method namely Simulated Annealing was applied to this problem. A new heuristic method (Tentacle) was also developed during this study and the results for the two methods are compared with results from two other studies on the same problem. Evaluation of all of the results in this study is based on volume recovery – that is the product volume after drying, edging, and trimming as a percentage of the original log volume. The reason to use volume recovery is its universal application and independence of specific regional pricing policies.

The objective of this study was to develop an effective and efficient search algorithm for finding optimal or near-to-optimal log positioning coordinates during the primary breakdown process in a sawmill.

2. Background

2.1 Economics of a sawmill

Sawmill volume recovery and optimisation thereof is a central concept in this study. The importance of optimising volume recovery can be explained at the hand of the following example. The important factors in the economy of a sawmill will be illustrated.

Assume there is a mill with the following parameters:

- The mill can process 200 cubic meters of logs monthly
- The mill's monthly overhead expense is R 100 000
- The mill purchases logs at R 500 per cubic meter
- The mill sells sawn timber for R 3 000 per cubic meter
- The current average volume recovery rate is 45%

Further assume the sawmill manager wants to increase the profitability of his mill. He has exhausted all options to reduce costs, his mill is running at full capacity but the market is growing and he wants to increase his sales.

He is evaluating two options: Firstly, invest in scanning and optimisation hardware to increase the average recovery from 45% to 49% or secondly, invest in an additional production line to increase his production capacity by 10%. In practice the recovery of timber was shown to increase by 4% - 5% by the use of optimisation software (Blackman, 1999).

A 4% increase in recovery may sound negligible and the 10% increase in capacity may seem more attractive but as can be seen from the basic economic breakdown, tabled below, it is not necessarily the case. In the table below the effect on the bottom line of the sawmill by either increasing recovery or process capacity can be seen. It should be noted that this example is rather simplified and only to illustrate the important factors in the economy of a sawmill.

Table 2: Basic sawmill economics – hypothetical mill example where a 4% increase in volume recovery is compared to a 10% increase in production capacity.

Item	Unit	Remark	Operating profitabilty of mill before changes	Option 1: Increase recovery	Option 2: Increase production capacity
Lumber volume sold	cub.m		90	98.0	99.0
Lumber price	R/cub.m.		R 3,000.00	R 3,000.00	R 3,000.00
Sales		Lumber volume sold x Lumber price	R 270,000.00	R 294,000.00	R 297,000.00
Log price	R/cub.m.		R 500.00	R 500.00	R 500.00
Recovery	%		45%	49%	45%
Logs processed	cub.m	Lumber volume sold divided by recovery %	200	200	220
Cost of Sale		Log price x log volume processed	R 100,000.00	R 100,000.00	R 110,000.00
Gross Profit		Sales - Cost of sales	R 170,000.00	R 194,000.00	R 187,000.00
Overhead expences			R 100,000.00	R 100,000.00	R 100,000.00
Net Profit		Gross Profit minus Overhead expenses	R 70,000.00	R 94,000.00	R 87,000.00

Increase in processing capacity	0%	10%
Increase in recovery	4%	0%
Increase in sales	9%	10%
Increase in net profit	34%	24%

It can be seen in this particular example that, assuming the two options cost the same, that the mill manager would get a shorter payback period from purchasing the optimisation hardware. The increase in net profit from a volume recovery increase is 34% compared to an increase of 24% when improving throughput.

It could further be noted that this example not only shows the importance of optimising volume recovery in a sawmill; but it also hints at the importance of your processing capacity. If your optimisation technique jeopardises this processing capacity it will be counterproductive. This study in essence evaluates techniques of optimising recovery by computer simulation but within limited time spans. An optimisation algorithm should be evaluated on effectiveness and efficiency (Reinders, 1989).

Where effectiveness is basically how close the algorithm gets to the optimum and efficiency how quickly it gets there.

2.2 Possible application for log positioning optimisation in a sawmill

This study forms part of an attempt to develop a log positioning optimisation system for typical South African framesaw sawmills. The complete concept is described in Appendix A. Although the initial idea was to develop a system for medium-sized framesaw sawmills, meta-heuristic search algorithms might also find application in modern automated log positioning systems. In such a system a log is scanned with a 3-dimensional laser scanner while the log is being moved on a conveyor chain - usually sharp chains which keep the log relatively stable. After the 3-dimensional scanning, the scanning data is used by a positioning algorithm to determine the optimal log position while the log is still moving between the scanner and a log rotation device. By the time the log reaches the rotation device, a solution must have been generated by the positioning algorithm. The rotation device then rotates the log before the alignment and skewing take place in front of the primary breakdown saw. The idea is that a meta-heuristic search algorithm may be used as the positioning algorithm in such a system.

3. Literature review

In this section literature is reviewed covering the most important aspects in this study e.g. the supply chain of a typical softwood mill in South Africa, different breakdown methods, volume recovery optimisation, log scanning and positioning systems, and combinatorial optimisation and heuristics.

3.1 Supply chain of typical softwood mill in South Africa

The plantation softwoods processed in South Africa is collectively known as SA Pine, which includes the five species Pinus patula, P. taeda, P. elliotti, P. radiata, and P. pinaster. Hardwoods used in South Africa include Eucalyptus and wattle species and are mostly used as raw material for the pulp and board mills. Softwoods are mostly used for dimension timber. This is sawn timber which is processed into standard sizes and which is commonly used for construction and manufacture of solid wood products.

The typical supply chain for a South African softwood mill can be broken down into the following sections.

3.1.1 Commercial tree growth in plantations

In South Africa almost all closed canopy natural forests are protected from active commercial use and almost all logs entering South African mills are sourced from plantations. As a matter of interest the total area of SA Pine Plantations in South Africa in 2009 was 650 183 ha from a total area under plantation of 1 274 869 ha (Godsmark, 2010).

The management of plantations and the silvicultural regimes including tree spacing, fertilization, thinning and pruning affects the quality of the final product of a sawmill substantially and the importance thereof should be appreciated. The focus of this study, however, is further down the value chain of wood and no attention is afforded to these aspects in this thesis.

3.1.2 Harvesting

When trees are cut down or felled the logs are cleared of branches. Bucking also takes place where the fallen trees are cut into logs. The final product length for which the timber is destined is of importance during bucking. Other factors such as sweep and defect removal are also taken into consideration during bucking and complex automated bucking optimisation systems are available (Maness et al. 1991, Faaland et al. 1984, Mendoza et al. 1986, Wang et al. 1989). These systems are however, not in general use in South Africa where manual bucking is the norm. As an example it should be noted that the sweep, a variable having a large impact on volume recovery, in a long log could be reduced to roughly a quarter of the original if the log is halved in length. Bucking could also be done in the log yard if full length trees are extracted from the forest.

3.1.3 Log yard

The logs are transported from the plantations to the mill where the logs are stored in the log yard. Logs are typically sorted according to the log class (sorting can also take place after debarking). The class of a log depends on the diameter but characteristics such as sweep, length and absence of certain defects could also be incorporated.

Some key characteristics of a log are illustrated in Figure 1.

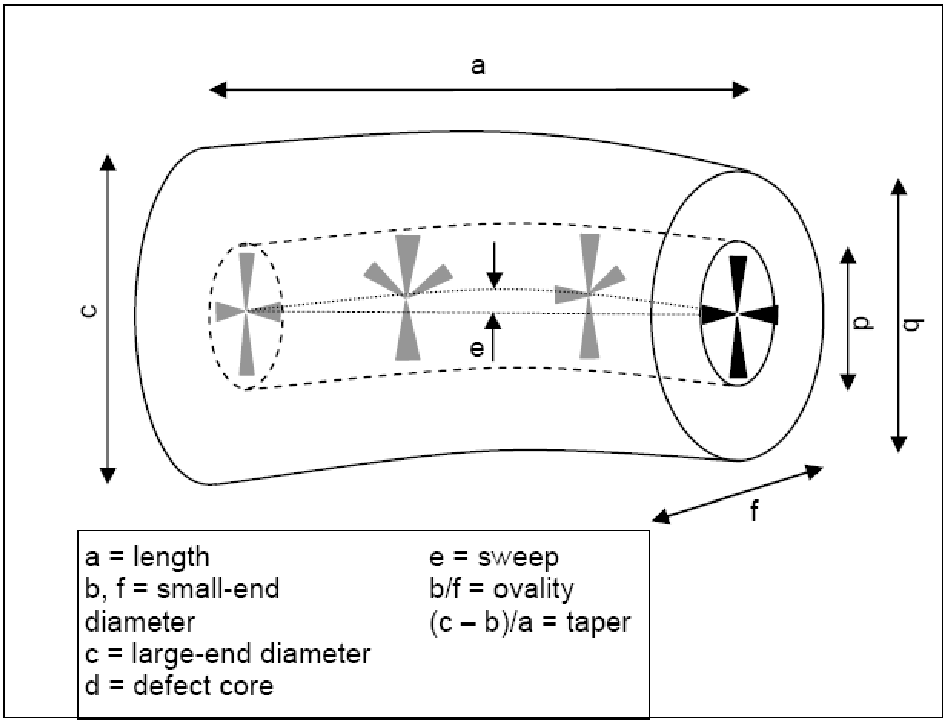


Figure 1: Log dimensional, shape and quality characteristics (Wessels, 2009a)

Log classes are often created in a way that specific sawing patterns can be used, which yield better results for that specific class (Wessels, 2009a). For example a log class of 36 cm diameter logs will be broken down differently than a log class of 18 cm diameter logs.

3.1.4 Debarking

Debarking as the name indicated is where logs are stripped from their bark. Various machines and processes are available.

The benefits of debarking will be explained briefly as it is of relevance in this study. The following are adapted from Bowyer et al. (2003), Denig (1993), Williston (1988) and Wessels (2009a):

Log evaluation is one the main benefits of debarking as modern scanning and optimisation equipment need images of a debarked surface to make optimal processing decisions. For primary log breakdown optimisation systems it is especially important that the log surface is without bark and loose pieces of wood strips. The profile that is sent from the scanner to the optimisation software is considered to be only solid timber and having bark or other pieces attached to the log may result in sub-optimal results from the optimisation software. Even in mills without scanning equipment, the sawyer can make better decisions on the breaking down of a log if he can see defects of the log – especially with grade sawing of hardwood logs.

Debarking also adds to extended tool life, less cleanup and maintenance, improved chip quality and the removed bark can also be sold as a by-product.

3.1.5 Primary Log breakdown:

Primary log breakdown can be described as the first sawing operation in the log's longitudinal axial direction that occurs in the sawmill (Wessels, 2009a). The primary breakdown together with secondary breakdown (described later) takes place in the wet mill. The primary breakdown saw must be able to handle a round log and is sometimes known as the headrig.

The primary breakdown machining process consists of the following steps:

- Log feed;
- Log evaluation;
- Log orientation or positioning;
- Log sawing;
- Out feed.

3.1.6 Secondary Log Breakdown

Following primary breakdown, secondary breakdown is the sawing operation where the remaining large pieces of the log are broken down to their final thickness dimensions (Bowyer et al. 2003). With cant sawing the cant will be cut into boards in the secondary saw. Some primary saws, like frame saws make secondary breakdown necessary but some sawmills do not actually need secondary log breakdown saws. The main aim of secondary breakdown, according to Denig (1993) is to free up the primary breakdown machine to increase production. Re-sawing can also be described as part of the secondary breakdown process.

More detailed descriptions of the log breakdown methods follows in Section 3.2.

3.1.7 Re-sawing, edging, and trimming

Re-sawing is when flitches, (i.e. boards with one face still in half round shape of log) are returned for sawing to thickness and where the half round face is removed. This is typically done with a horizontal band saw or circular saw so that the flat face can lie on the saw table.

Trimming on the other hand is where a board is cut shorter to remove wane and to set the product length. Wane is the appearance of bark typically on the edge of a board. Depending on the grade of sawn timber the tolerance for wane varies. Dimension timber, like SA Pine, is only cut to specific lengths. The mill sells boards in bales of around 1 cubic meter and from the mill's perspective it would be impractical to stock too many product sizes.

Edging is done to remove irregular sides containing wane and bark leaving four sided sawn timber.

Value recovery of sawn timber is affected greatly by trimming and edging and optimisation hardware and software are commonplace in secondary breakdown of logs.

3.1.8 Drymill

The drymill refers to the processes needed during and after the drying stage to get final saleable products. These processes will differ from mill to mill but may include:

- Drying
- Grading

- Grade optimisation
- Finger jointing
- Lamination

All these processes have the need for optimisation and the profitability of the mill is very dependent on the efficient operation of these plant areas. It is however considered to be outside the scope of this study.

In the next section the different breakdown methods of logs into sawn timber is discussed.

3.2 Different breakdown methods in a sawmill

There are many ways in which a log can be broken down into sawn timber. Most methods can be categorized as one of the following: live sawing, cant sawing, grade sawing or radial sawing (Walker, 1993). In South Africa a fifth method namely growth stress breakdown is used for breaking down eucalyptus logs. Each method has different applications and advantages. The selection of a breakdown method is affected by considerations such as the type of timber used, end product properties required, log dimensions and the throughput required. Another important factor in selecting a breakdown method is which grain orientation is needed in the final product. As per example, radial orientated grain on the broad face of boards in the teak used in the marine industry is critical. The value of this sawn timber drops so dramatically if this grain is not present that volume recovery is almost of no importance.

These methods will be explained briefly as reference is made to some of them.

3.2.1 Live sawing

In live sawing the whole log is sawn into boards with the primary saw and all cuts are parallel to each other, see Fig 2 (Walker, 1993). The boards are mostly dried before further processing. This method is suitable for grade recovery as the subsequent cuts are made from wide slabs and the cutting can be optimised. Warping and cupping of the final product is also reduced as it is cut from dried slabs. The primary saw would need a big cut depth in this method and frame and band saws are ideal.

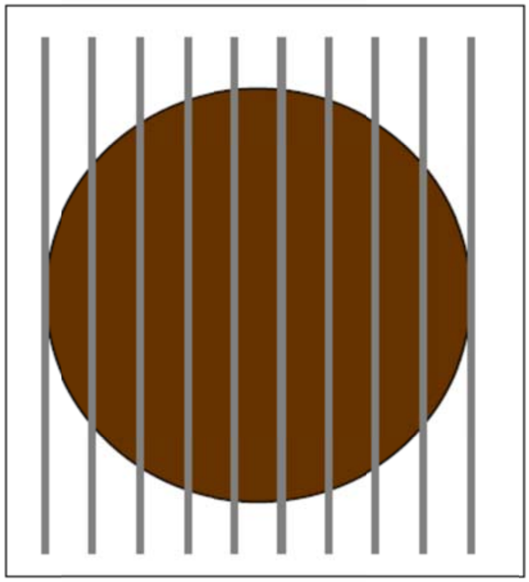


Figure 2: Live sawing method

3.2.2 Cant sawing

In cant sawing two slabs are firstly removed from either side of the log. The remaining piece, which is called a cant, is then rotated 90 degrees before it is sawn, typically by a secondary saw, into boards. This method is popular in South African softwood mills due to good volume recovery and the processing speed possible. The saws used can include band saws, frame saws, circular saws and chipper canters. The recovery in this method is especially applicable if round-the-curve sawing is practiced (Carino et al. 2006, Hamner et al. 2005). When a log with considerable sweep, i.e. being shaped like a banana, is rotated so that the ends point upwards, this position is referred to as the horns-up position. When a cant is sawn from a log in this position it will, when lying on one face, have a curve in the horizontal plane. If the secondary saw can follow this curve it should be evident that the resulting volume in boards will be greater that if straight cuts were made. The boards will obviously have a banana shape but this will be corrected during drying. See Figure 3 below for a schematic of this breakdown method.

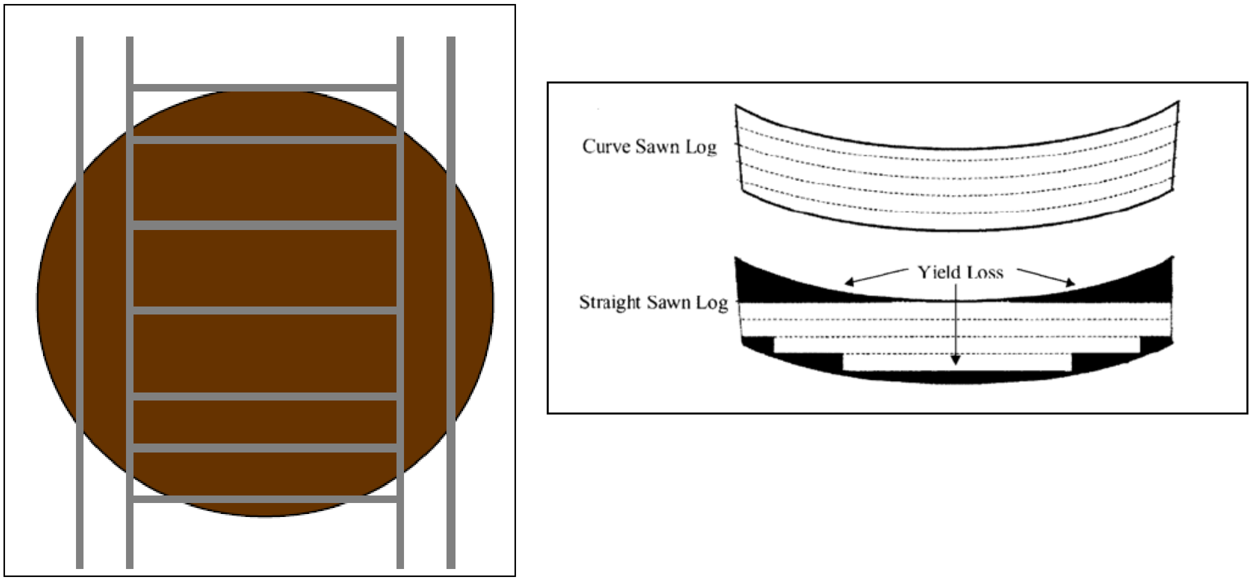


Fig. 3: Cant sawing method showing the advantage of curve sawing (Hamner et al., 2005)

3.2.3 Grade sawing

Grade sawing is also aptly referred to as sawing around, considering the following description. Firstly slabs will be sawn from the one side of the log until the sawyer decides to turn the log 90 degrees. He will then continue to saw slabs from the next side and repeat the process, (thus sawing around) until all timber is recovered. The sawyer’s decisions of which side to start with and when to rotate the log is based on maximum value recovery. One factor to consider would be to avoid the knotty core in pruned logs and maximizing the recovery of clear sawn timber. This method is mostly used for high value timber in large diameter logs (it is therefore of little importance in South Africa). The importance of the sawyer’s skill and experience with this breakdown is decisive. A carriage bandsaw or mobile bandsaw can be used for this method.

See figure 4 which gives a schematic description of this method.

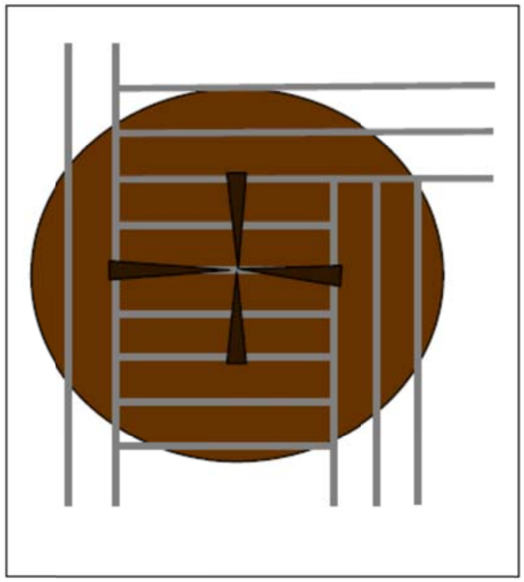


Fig. 4: Grade sawing method

3.2.4 Radial sawing method

Radial sawing is a very slow process and the main benefit is the production of products with radial grain. An example for the need for radial grain is referred to under heading 3.2. Various methods are grouped under radial sawing, see figure 5 for a schematic presentation of two of these methods.

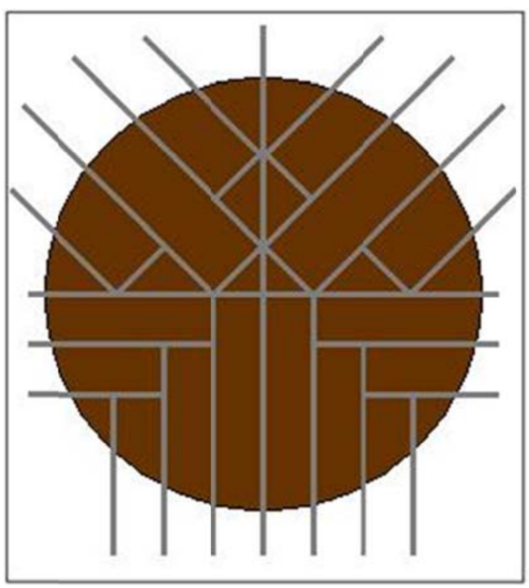


Fig 5: Radial sawing

3.2.5 Growth stress breakdown

Some trees contain substantial growth stress. For example eucalyptus trees grow rapidly which contributes to the centre to be in compression and the exterior in tension (Malan, 1988). If the boards are removed from the log asymmetrically the remainder of the log will bend severely because of this latent stress inside the log. This will increase the difficulty in processing logs as most processing equipment has limits to the amount of sweep it can handle. The objective in this method is therefore to firstly remove boards from the log symmetrically but also to have boards with only one of the stress regions present. If a eucalyptus log was live sawn, for instance, the boards in the centre of the log will have tension on the edges and compression in the centre. As these incorrectly sawn boards leave the primary saw they will rip themselves into two banana shaped boards that will have almost no use. See figure 6 below, in which this method is presented schematically.

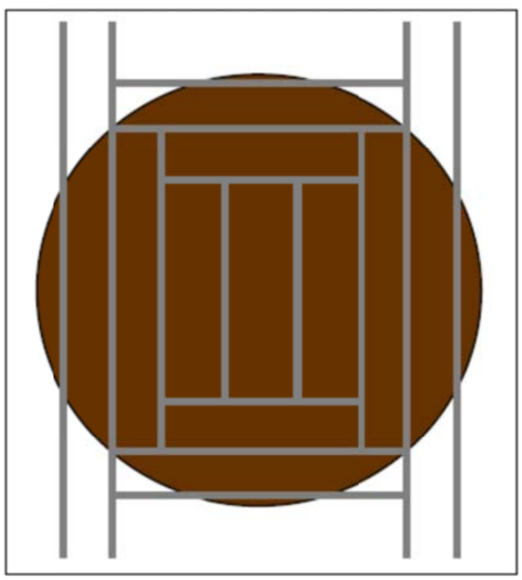


Fig 6: Growth stress breakdown

3.2.6 Log positional variables

In this study log orientation or positioning before primary breakdown are varied to find an optimised position in terms of log volume recovery. The definitions of these variables are illustrated in figure 7 below. All definitions are illustrated with two instances, firstly with the neutral, centered position (usually horns up position) and then with the new orientation illustrating the particular definition. The broken line indicates the neutral, centered feeding direction of the log into the saw. The solid line indicates the central axis of the log.

Offset is the distance by which the central axis of the log is moved parallel to the neutral feeding direction of the log;

Skewing or aligning is the distance by which the one end of the log is moved, while keeping the other end stationary. This creates a new angle with which the log will enter the saw.

The rotation is defined as the rotation of the log around its own axis.

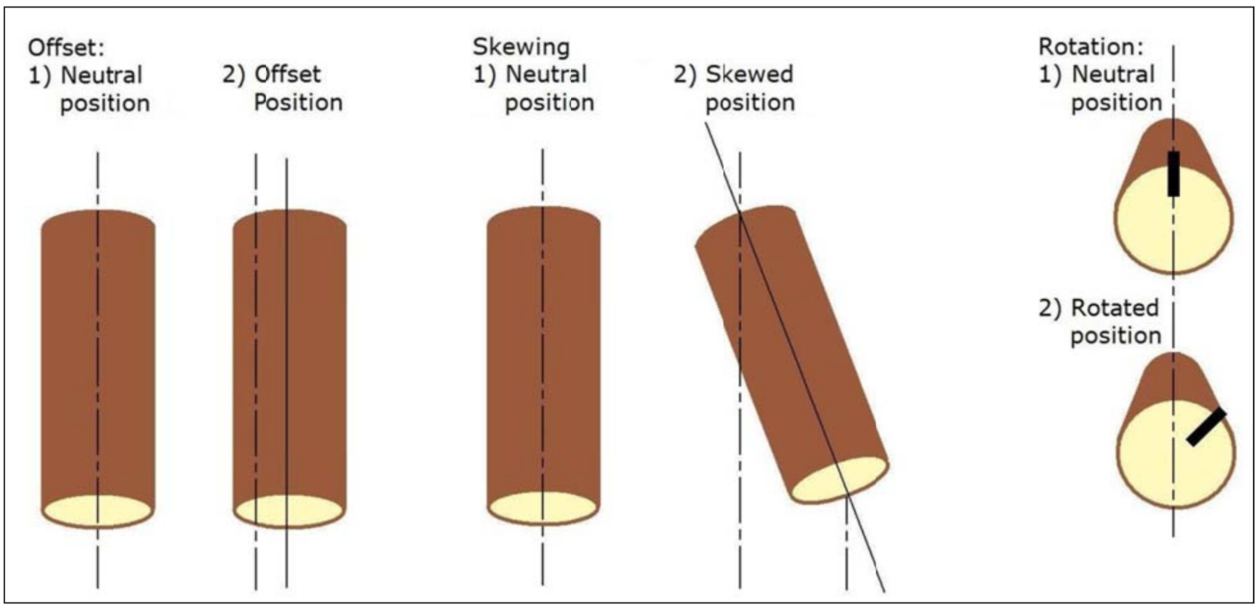


Fig. 7: Definition of log positioning variables

3.3 Log scanning and positioning systems

3.3.1 Scanning

When a log is evaluated for positioning a digital image of the log is required. In a production mill there will not be sufficient time to do manual measurements and scanners are used to digitise the characteristics of each log. In obtaining a digital image of the exterior of a log, two types of scanners are commercially available:

Shadow scanners:

In this case a row of LED's is positioned across a row of light sensors. Each LED emits a beam of light which is picked up by its facing light sensor. When an object moves through this curtain of light the LED's will be obscured from the light sensors. Some information can be gathered regarding the size and shape of the object depending on which LED's are obscured. It should be evident that the size of the object is based on the shadow which is cast by the object. Looking at one's own shadow it is evident that little information can be gathered of the three dimensional shape of one's body. To increase measured accuracy, the pairs of LED and light sensor rows can be increased to measure multiple axis of the object. This type of scanner is however still only used to measure the length, diameter and taper of logs due to the low resolution. For a single-axis shadow scanner only two points per cross-section on a single axis can be determined. For a two-axis shadow scanner four points or two points per cross-section can be determined. That is a rather low resolution.

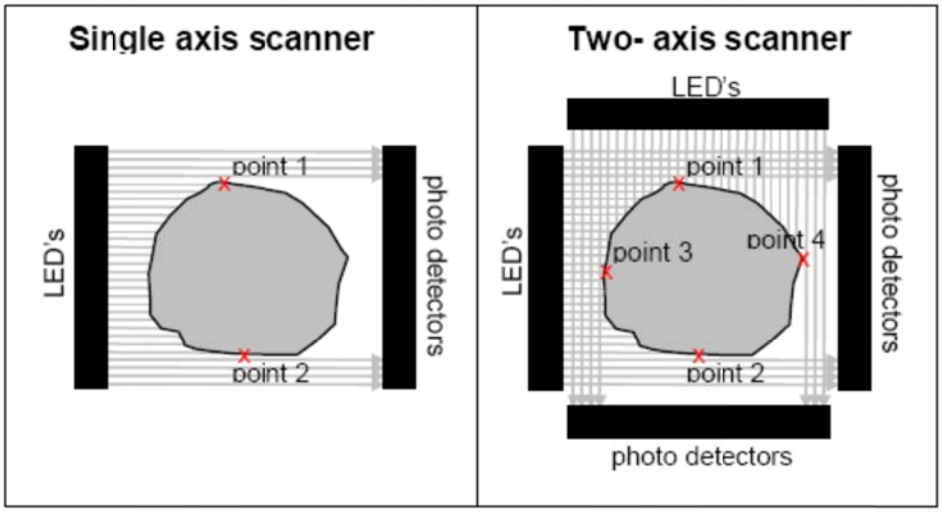


Fig 8: Single axis and two-axis shadow scanners provide position data on two points or four points per cross-section respectively for continuous feed systems. (Wessels, 2009a)

Laser scanners:

Laser scanners take digital photos of laser lines that are shone on an object. The exact positions where the laser meets the body can then be determined through triangulation. When four cameras for instance are evenly spaced around a log, the dimensions of the log's outer perimeter can be very accurately measured with these cameras. When these images are generated at regular intervals along the length of a log a three-dimensional image of the log is possible. This is the hardware that is required to generate high resolution digital images of logs which are needed in positioning optimisation. Resolution of these log images are far superior to shadow scanners and can be in the region of 0,1 x 2 mm^2 with an accuracy of approximately 0,1 mm depending on measurement range (Thomas et al. 2004).

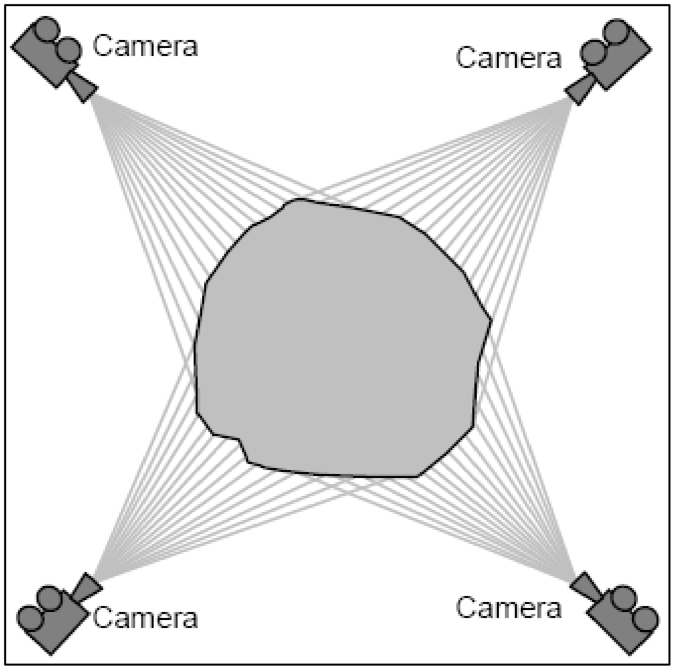


Fig 9: Three dimensional laser scanning provides high resolution images of a log. (Wessels, 2009a)

3.3.2 Log positioning and feeding

When looking at optimal positioning of logs one has to consider the sensitivity of these 'optimal' results with regards to positioning errors. It will be counterproductive to find the optimal position for a specific log only to break the log down in a different position. A study was performed by Wessels (2009b) to estimate the sensitivity of positioning errors - see the table below for the results.

Table 3: Estimated average loss in volume recovery for different levels of positioning errors. (Wessels, 2009b)

Error description	Level of error	Offset (%)	Skewing (%)	Rotation (%)	Combination of errors (%)
Low error	+/- 2mm or +/- 2,5 degrees	1.34	0.77	0.77	1.06
Medium error	+/- 5mm or +/- 7,5 degrees	2.40	1.22	1.49	2.11
High error	+/- 10mm or +/- 10 degrees	3.50	1.96	2.31	3.39

The results in the table above makes it evident that the potential benefit of optimising the position of a log can be lost when the equipment used to position and feed a log into the saws are not accurate. Another study (Vuorilehto and Tulokas. 2007) has found that positioning errors in practice can be substantial. They found errors in rotation of above 20 degrees.

Feeding and positioning systems used in the industry are suitable for either continuous or carriage feed. In continuous fed systems positioning is done by either a log carriage or a rotating tube system. With the log carriage system the log is held by a front and rear carriage that can move independently. This allows for the offset and skewing to be adjusted. Rotation of the log is done by hydraulic clamps. Once the log is fed into the saw, two rollers usually hold the log in position. With the rolling tube system the log will lie parallel and on top of two long adjacent cylindrical rollers. The rollers will turn around its own axis causing the log to rotate to the required position. The rollers can also move laterally to change the offset of the log. With this configuration no skewing of the log is possible. A chain running between the cylinders together with an end-dog will feed the log into the saw as soon as it is in the desired position.

Once in position, the log is fed into the saw by a sharp chain, lug chain or overhead carriage. The sharp chain is the most popular option (Williston, 1988). The log lies on a chain that runs in the feeding direction of the saw. Sharp protrusions on the chain bite into the log. Rollers are also utilized to hold the log down onto the chain. Lug chains use lugs (sometimes referred to as dogs); which protrude from the chain to push the log into the saw. This method provides less stability than the sharp chain for the feeding of the log. Lastly an overhead carriage can be used. This method provides the most lateral stability for feeding of a log (Williston, 1988). The overhead carriage consists of two hydraulic arms gripping the log on the two opposing cut faces. The grips (or dogs) are narrow enough to feed the log through the cant saw. This method is not suitable for small diameter logs.

Carriage feed or single cut systems uses a wheeled carriage that runs on rails (Walker, 1993). See picture below. The carriage has at least two independently movable 'knees'. This allows changing the offset and depth of cut. The dogs hold the log in place. This method is very flexible and more suitable for large diameter valuable timber as it is not suitable for high volume production. It is also a very accurate feeding system.

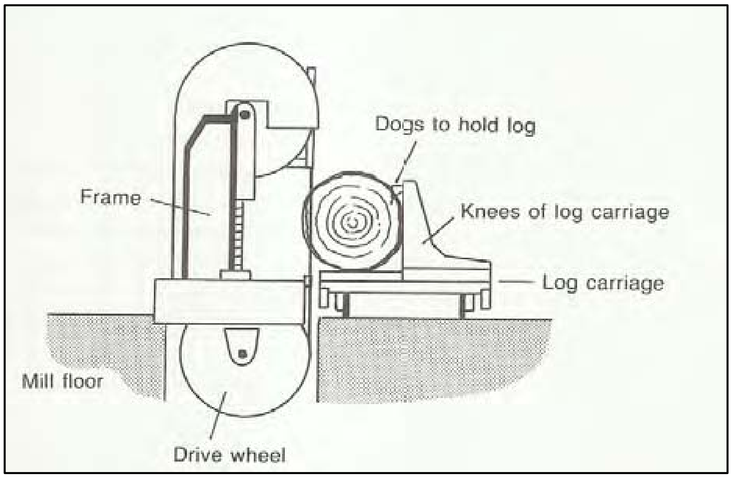


Fig. 10: A log carriage for a single cut bandsaw system (Walker, 1993).

3.4 Sawn timber recovery optimisation

With the importance of sawn timber recovery clearly illustrated in the introduction it will be no surprise that prolific research has been done on the topic. Since the 1960's researchers have looked at this problem from various angles. With the advance in computer systems the methods for increasing recovery grew in complexity. The research methods used to date can be crudely categorised under the following headings:

- Empirical, where real logs were sawn using a number of processes and where the processes were evaluated;
- Theoretical, where the recovery is calculated mathematically;

- Simulation studies, where the sawing process is simulated using artificially created or scanned images of real logs and recovery is calculated based on different input variables.

Some of the relevant research and findings will now be discussed under these headings. It is important to know the ground work for this particular study. Some key concepts will also be explained.

3.4.1 Empirical

Some studies were done where logs were batch sawn, i.e. a number of logs are grouped and all of them are sawn to one particular pattern. Cown et al. (1988) sawed 275 logs with the use of 7 different breakdown methods. The researchers found that the logs positioning "…appeared to influence conversion slightly." They also found that exterior defects and silvicultural records are not always good indicators for value recovery.

The benefit of curve sawing was also the focus of an earlier empirical study. Wang et al. (1992) estimated the increase in recovery as much as 16% in 11cm top diameter logs. It should be noted that the benefit of curve sawing is not questioned in this present study. Curve sawing is, however, used in the secondary breakdown of the simulations done. As a matter of interest the study obtained a regression equation to describe the dependency of recovery on the various factors including: diameter, taper, sweep and rotation.

The advantage of empirical studies is that results are the most trustworthy – no simplified models are used. The disadvantage of such an approach is that logs can only be sawn once, and large numbers of logs have to be used to statistically reveal certain effects based on averages and variances. The sawing patterns are also limited to the technology available in the production line.

3.4.2 Theoretical (mathematically calculated)

The problem of optimising recovery in the sawing process has also been looked at from a purely theoretical point of view. The disadvantage of a theoretical approach is that the log shape is often reduced to an abstract body of revolution. The advantage is that if the log's form description is accurate, volume optimisation can be achieved.

Maximizing the yield of sawn timber through the sawing of a log can be considered to be a packing problem; it is also sometimes referred to as a knapsack problem. In other words, the problem is the determination of the maximum number of sawn timber of set sizes that can be packed into a log of a certain size. Reinders et al (1989) published a paper in which an algorithm was developed through which the recovery of sawn timber could be maximized. It is of interest that the computer time needed to run the model in the study was as much as 88 seconds. Computer technology has obviously advanced in the last 20 years but the possibility to lose valuable time in the mill should be appreciated.

Another study by Geerts (1984) was done where a dynamic programming algorithm was derived. In this study the computer time needed to obtain a solution in one instance was 245 minutes.

Without detracting any value of the above two papers the author of the present study wishes to highlight some of the differences in the approach of their studies to the present study. Firstly, to solve this problem from a mathematical point of view the authors simplified the problem somewhat by assuming the log is perfectly round and also that the sawn timber recovered comes from a cylinder without taper. In the first study the effect of kerf was also omitted, which could arguably be added without too much difficulty but Pinto et al. (2006) found that volume recovery can drop by as much as 3% with 1mm additional kerf. It is therefore argued that omitting kerf could cause incorrect results. To assume a log to be perfectly round is even more problematic. The empirical study by Wang et al. (1992) showed that curve-sawing increased the volume recovery by up to 16%. This increase is purely due to the sweep or curvature present in logs. Omitting this variable will thus result in fairly large errors – especially for trees which have large sweep and taper values.

Carnieri et. al (1994) also presented algorithms addressing the knapsack problem of optimising the cutting of dimension timber or particle board.

The primary and secondary breakdown of a log is of a sequential nature and to attain a global optimal recovery they need to be optimised simultaneously (Todoroki and Ronnqvist, 1999). This study linked these two breakdown steps in a dynamic programming problem to maximize the global recovery. The study also used simulations of 40 logs to illustrate the use of their methodology. They focused on live sawing and no possibility of skewing. The authors mentioned that heuristics might be needed when skewing is included as a variable.

Many optimisation algorithms is focused on a population of logs (i.e. log yard inventory) and also takes into account market constraints (i.e. Bryan, 1996; Todoroki and Rönnqvist, 2002; Wessels et al., 2006). This study however, assumes that the market constraints has been taken care of through the development of sawing patterns and only focuses on positioning of the log.

3.4.3 Simulation

Lastly we look at the work that was done where the outset of the study was based on simulation of the sawing process. The selection of optimum sawing patterns and positioning methods for logs using simulation software has been studied extensively since the 1960's by several workers (e.g. Peter and Bamping, 1962; McAdoo, 1969; Hallock and Lewis, 1971; Steele et al., 1987; Maness and Donald, 1994; Chang et al., 2005). With the availability of sawmill simulation software, logs can be virtually sawn an infinite number of times with different methods and the outcome of each can be evaluated before one needs to decide which method to use on the actual log. The optimum is therefore decided upon, by an iterative process. Due to the number of studies only a selected few will be discussed here.

Barbour et al. (2003) conducted a study where a simulation software package was used to evaluate a number of sawing patterns for small diameter logs. In this study they also followed the simulation study with an empirical study and found that the software that they used estimated the recovery 10% – 15% shy of the actual recoveries.

Another very well known method for maximizing recovery through an iterative process is the Best Opening Face method (Lewis, 1985; Steele et al. 1987). In this method the log scanner determines the diameter, taper and length of the log. The model first determines the opening face that will produce the smallest acceptable piece of timber. The successive cuts can then be made and the resultant recovery determined. This process is repeated but with the opening faces incrementally moving towards the centre of the log. When all the reasonable possibilities are examined the Best Opening Face is chosen. It should be noted that log sweep is considered as zero in this approach as the log is assumed to be perfectly cylindrical. The effect of log rotation is therefore not tested in this model.

Subsequent studies have shown that log rotation does influence recovery significantly. Maness et al (1994) showed that the horns-up position in general outperforms other rotations of the log.

The BOF procedure was also used in another simulation study where 8 specific breakdown strategies where compared (Hallock et al, 1976). 3510 different log combinations were "computer sawn" with the different strategies. The most suitable strategy for each log class was determined. In this study the logs were theoretical, i.e. not scanned or measured from real life logs. The benefit of simulation is, however, highlighted by the fact that 3510 logs could be "sawn" without any material cost.

In another study the effect of sweep was analysed using computer images of real logs but with varying degrees of alteration (Monserud et al 2004). This illustrates another use of simulation where hypothetical sawing scenarios can be created by editing real life computer images. This particular study found a decrease of 10% in recovery for each additional 100 mm increase in sweep.

Wessels (2009b) conducted a simulation study on log positioning. The researcher explored reduced range sizes in log positioning within which near optimal results can be still be found. He also considered the increment sizes adequate for a simulation study and the effects of errors in log positioning before breakdown.

Chiorescu and Grönlund (2000) conducted a study in which grade sawing was simulated. For this study they used CT (computer tomography) scans of 625 logs. This allowed them computer models of all the internal defects of the logs. Some of the concepts behind the computer modelling of logs together with its internal defects were presented by Occena and Tanchoco (1988). The simulated results in the Chiorescu study were also compared to the actual output of the sawmill. They concluded that the simulated results gave a good indication of what to expect from the actual mill. Although that study is ahead of its time regarding the technology available in most sawmills, it gives an interesting perspective on what the future holds for sawmill optimisation. It should also be noted that grade sawing is not considered in the present focus on volume recovery.

The advantage of simulation studies is that logs can be sawn infinite times, real data can be used as opposed to mathematical optimisation, and log abstraction can be small. The disadvantage is that it is a time consuming process (iterative) and some simplification to the real process still exists.

3.5 Combinatorial optimisation and heuristics

The previous chapter highlighted techniques that has been used to analyse or optimise volume recovery in sawmills. Most of these techniques are not suitable for individual log positioning. To reduce the cost (or time) other methods can be used such as combinatorial optimisation and heuristics. As explained earlier, combinatorial optimisation problems are problems where the set of feasible solutions is discrete (or can be changed to be discrete), and the goal is to find the best possible solution. The challenge with combinatorial optimisation is that the number of possible combinations grows at an alarming rate with an increase of variables and variable ranges. For example, the problem of log positioning can become quite big if the following combinations are considered:

- Firstly for log offset alone; where the offset can range between -30 mm and +30mm with an increment of 1 mm. In this instance 61 positions need to be evaluated.
- When we add log skewing as a variable, which can range between -30mm and +30 mm with an increment of 1 mm we need to evaluate 61 x 61 = 3 721 combinations.
- When we add log rotation as a variable, which can range from 0 to 359 degrees with an increment of 1 degree, we need to evaluate 61 x 61 x 360 = 1 339 560 combinations.
- When we add cant offset as a variable, which can range from -10mm to 10 mm with an increment of 1mm, we need to evaluate 61 x 61 x 360 x 21 = 28 130 760 combinations
- Etc.

In this example the different combinations of log and cant positions can be referred to as the search space. If every combination is evaluated to attain the optimum combination an exhaustive search is conducted. It should be evident that an exhaustive search in optimisation has its limitations even with the power of today’s computers. In this study each combination must be simulated on sawmill simulation software. This makes the computation cost even higher.

To be practical one can sacrifice the need to get the absolute optimal solution to a problem and to rather find a near optimal solution at a fraction of the cost (i.e. time). We can use heuristics with this approach. The term heuristic is derived from the Greek word for “find” which can be defined as follows (Rayward-Smith et al. 1996):

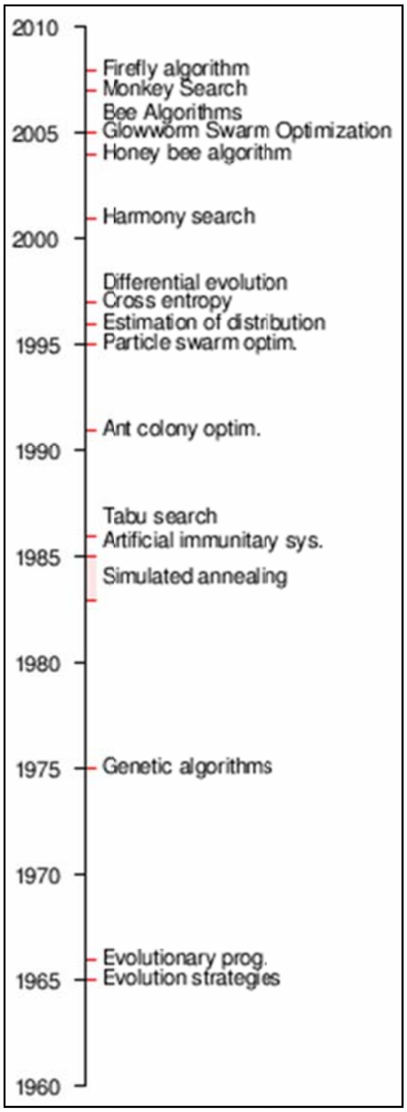
A heuristic technique (or simply, a heuristic) is a method, which seeks good (i.e. near optimal) solutions at reasonable computational cost without being able to guarantee optimality, and possibly not *feasibility. Unfortunately it may not even be possible to state how close to optimality a particular heuristic solution is.*

An advantage of using a heuristic is that fewer simplifying assumptions are needed as opposed to an exact mathematical approach.

Numerous heuristics and meta-heuristics have been developed. A meta-heuristic can be described as a higher-level general strategy which guides other heuristics to search for a feasible solution. Some of these meta-heuristics have imitated natural phenomena in their search for a near optimal solution.

See the table below for a timeline of when certain well known meta-heuristics have been developed [www.wikipedia.org].

Table 4: Timeline of development of certain meta-heuristics (www.wikipedia.org)



As mentioned earlier an exhaustive search will lead to a global optimum, i.e. the optimum for the entire search space. In contrast a global optimum is a local optimum, where a local optimum is the optimal solution of some smaller collection of possible combinations in the search space, typically referred to as a neighbourhood (Rayward-Smith, 1996). An analogy of a man walking up a hill in a dense fog looking for the summit can be used to illustrate this concept (the hill climbing method is a heuristic that is commonly used.) This man will only be able to see the area immediately surrounding him, i.e. the points neighbouring his current location (i.e. focal point). He cannot see the complete mountain range so the global optimum is not known. He can evaluate the elevation of the points in his 'neighbourhood' and decide where to place his next step. In doing so his focus jumps to a new point on the hill. By repeating these actions the man will reach a point, sooner or later, where all the points in his neighbourhood have a lower elevation; thus he reached the top of a hill. This hill can be referred to as a local optimum as there is no way of knowing whether this is the summit of the whole mountain range. Most meta-heuristics try to escape these local optima, some by allowing the focus to move to poorer combinations in the hope that the meta-heuristic will eventually reach stronger combinations, i.e. higher hills or even the summit. All meta-heuristics aim to cover the whole search space effectively and with a high efficiency in avoiding local optima.

One such meta-heuristic is the simulated annealing meta-heuristic which is a relatively popular technique and which is described below.

3.5.1 Simulated Annealing (SA)

This meta-heuristic is based on the metallurgical process of annealing. The ideas that form the basis of this heuristic were first published by Metropolis et al. (1953). In metallurgy the properties of a material can be manipulated by the controlled cooling of the molten material. Large crystals for instance can be grown inside a material with very slow cooling where quenching of the material will lead to small crystals with a number of imperfections. This process is known as annealing. It is argued that the slow cooling gives the atoms in a metal more time to move around and find configurations with lower internal energy than the initial one. In other words the atoms are freed from local optima.

This heuristic technique works by searching the set of possible solutions and avoiding getting stuck in a local optimum by allowing moves to inferior solutions under control of a randomized scheme. This move to an inferior solution will be accepted if:

$$\exp(-\Delta c / T) \geq R$$

Where:

Δc = Change in value of solution

T = A control parameter, in this case Temperature

R = A random number between 0 and 1

It should be evident that the move to inferior solutions will be very probable initially with a high value of T but will become less likely as T approaches 0. It should further be noted that the cooling schedule and the starting temperature are two important parameters in the effectiveness of this search method.

The structure of the simulated annealing algorithm is quoted from Michalewicz and Fogel (2004):

Simulated Annealing Algorithm:

procedure simulated annealing

begin

$t \leftarrow 0$

initialize T

select a current point *Va* at random

evaluate *Va*

repeat

repeat

select a new point v_n in the neighbourhood of v_c

if *eval* (v_c) < *eval* (v_n)

then $v_c \leftarrow v_n$

else if *random* [0,1) < $e^{(eval(v_n) - eval(v_c))/T}$

then $v_c \leftarrow v_n$

until (termination condition)

$T \leftarrow g(T, t)$

$t \leftarrow t + 1$

until (halting – criterion)

end

Variations of simulated annealing such as re-annealing has also been developed (Ingbar, 1989).

3.5.2 Population based incremental learning (PBIL)

PBIL is an algorithm that was developed by Shumeet Baluja in which he combined the mechanism of a generational genetic algorithm with simple competitive learning (Baluja 1994). It proved to be a simpler algorithm that out performed genetic algorithms in many instances. One great difference between Genetic Algorithms (GA) and algorithms such as Simulated Annealing (SA) is the concept of parallelism. In SA or in a classic hill climbing algorithm only one point in the function space, or schemata as referred to by Baluja (1994), is used as the basis of the search. With GA's and PBIL multiple schemata is searched in the solution vector; thus the schemata is searched in parallel. Stated differently, in PBIL (and GA) a population of solutions is created in each iteration and the suitability of each solution is evaluated. In SA for instance only one solution is generated and evaluated against the solution under consideration.

In the PBIL algorithm a probability vector is used as the basis for its search of an optimal solution contrary to GA's where the population is used as basis. In other words in GA's the actual 'DNA' of the population changes with subsequent populations but in PBIL only the probability of certain 'DNA' characteristic changes. The probability vector is an encoded set of the variables arranged in a linear array, and each member of the set, or cell in the vector contains a probability (Bekker, 2006). If we assume binary coding for the solution vector (for instance an 8 cell vector containing 0 or 1) the probability vector will have the same number as cells as the solution vector but each cell containing a probability that the cell of a solution vector will contain 1. The probability vector will start with a value of 0.5 for each cell. A number of solution vectors are generated from this probability vector. This collection of solutions is called a population, and the size of a population is a parameter that can be adjusted by the user to optimise the efficiency of the algorithm. The population of solutions is evaluated and the probability vector is trained towards the most suitable solution. This is done by the probability update rule, (Bekker, 2006):

$P(i) \leftarrow P(i) \times (1 - LR) + (SMax(i) \times LR)$

where

P(i)	=	Value of the i-th element in the probability vector
LR	=	Learning rate (typically 0.1)
SMax(i)	=	Value of the i-th element in the solution vector (0 or 1) yielding the maximum evaluation value

The learning rate adjusts the rate with which the algorithm converges to a solution. The higher this rate is, the smaller the portion of the search space that will be explored.

With repeated iterations the probability vector will converge to a single point where the cells in the probability vector have converged to either 0 or 1. This converged vector will be presented as the solution to the optimisation problem.

Another strategy of PBIL to avoid being caught in local optima is mutation. It is especially important in later populations when some of the diversity has been lost. Two parameters needed to define this aspect is mutation probability and mutation shift. After learning of the probability vector, during each iteration, each cell of the probability vector is subjected to mutation. The likelihood and degree of mutation is specified by the mutation probability and mutation shift.

Lastly, the PBIL algorithm has a set condition that will prevent the algorithm to run indefinitely and to terminate. The termination of the PBIL algorithm can be done after a certain amount of time or with thresholds being reached for each cell in the probability vector. Typically 0.05 or 0.95 is used as these thresholds.

3.5.3 Particle Swarm Optimisation (PSO)

Particle swarm optimisation is a population based stochastic optimisation technique developed by Kennedy and Eberhart (2001), inspired by the social flocking of birds or schooling of fish. The PSO philosophy of optimisation is that of a swarm of birds, in-flight, looking for a single location of food. The location of the food is not known but each bird's (or particle's) distance from the food is known and is communicated to the whole flock. The most effective solution to finding the food is argued to be the following of the closest position to the food.

In the PSO method particles are initially randomly placed in the search space moving in random directions (Pederson, 2009). The direction of the particles is then gradually changed, in each iteration, towards the previous best position of the particle and the previous best position of the whole population. Each position is evaluated with regard to some fitness measure, namely: $f : \mathrm{R}^n \rightarrow \mathrm{R}$.

Let the position of the particle be denoted by $\vec{x} \in \mathrm{R}^n$ and the velocity by $\vec{v}$. The particle's velocity is updated by the following equation (Shi 1998):

$$\vec{v} \leftarrow \omega\vec{v} + \phi_p r_p(\vec{p} - \vec{x}) + \phi_g r_g(\vec{g} - \vec{x})$$

Where

$\omega \in \mathrm{R}$ is called the inertia weight and controls the recurrence of the particle's velocity

$\vec{p}$ and $\vec{g}$ is the particle's and the swarm's best previous position respectively

r_p , $r_g \sim \cup\,(0,1)$ are stochastic variables weighing the user defined variables, i.e.,

$\phi_p, \phi_g \in \mathrm{R}$

It is customary to impose limitations on the velocity of a particle and each particle also needs to be bounded inside the search space.

The user defined variables ϕ_p, ϕ_g can be respectively referred to as the "memory" and "cooperation" of the particle (Del Valle, 2008). In other words each particle's behaviour can be adjusted to favour his own previous best position or the previous best position of the swarm.

In summary, in PSO a population of particles is randomly generated with a specific position and velocity. Each particle is then evaluated. Each particle's best position and the swarm's best position are recorded. The velocity of each particle is modified with the above equation. Adding the velocity of each particle to its position causes each particle to move to a new position in the search space, $\vec{x} \leftarrow \vec{x} + \vec{v}$. The process is repeated until a specified number of iterations have been completed or until a convergence threshold is reached.

4. Methodology

The basic steps in this study were the following:

1. 60 real logs were virtually recreated using a specially manufactured jig and measuring system.
2. Sawing of each virtual log was simulated in a large number of different primary breakdown positions. The offset, skewing and rotation of the logs were varied. The data set formed by each combination with its corresponding volume recovery, formed the search space within which each different meta-heuristics could be tested.
3. The efficiency of the SA, PBIL and PSO algorithms were evaluated in finding near optimal combinations within the search space.
4. Exhaustively searching of high quality areas in the search space was considered as an alternative.
5. Random selection of combinations was also considered as a search strategy and was used as benchmark for all search strategies.
6. In addition a new meta-heuristic was developed and was tested against the other methods described.

4.1 Simulation study

The simulation of logs formed part of a different study (see Wessels, 2009b) but the results were used in this study. The author was involved with the sawing simulation of the logs.

Simsaw 5R was used for the simulation of the log breakdown process. The first version of this computer software was developed by the CSIR in South Africa in 1975 and the latest version has the ability to simulate three-dimensional log images (Wessels et al. 2007). The software allows the user to import the spatial coordinates of the circumference of a log at various points along the length of a log. A three dimensional wire frame or digital image of the log is therefore created by the computer. This wire frame will portray the characteristics of the log, for example the sweep, ovality and taper. The software can then simulate the breakdown process after the user has defined various parameters.

The digital images of 60 real life logs were randomly selected from a log database for this simulation study. The database included logs of the species Pinus patula, Pinus elliotti and Pinus taeda all grown in the Mpumalanga region of South Africa. These real log images were created by digitizing sample logs using a mechanical measurement system together with a specially manufactured log jig to ensure a high degree of accuracy. The digitizing of a log involved loading it onto the jig and marking positions along the length of the log where a digital cross-sectional profile was required. The X-Y-Z coordinates of two points on each cross section were determined by measuring the distance from two reference lines on the jig to each point. Each cross section was then removed using a chain saw and the exact profile of the cross section digitized. The two marked reference points on each cross section provided the means to position the cross-sectional profile correctly in three-dimensional space and creating an accurate log image. The logs were sorted into 3 classes based on their small end diameter (SED):

- Class 1 for a diameter of 180 mm to 219 mm
- Class 2 for a diameter of 220 mm to 279 mm
- Class 3 for a diameter of 280 mm to 339 mm

The parameters for the software were selected to portray the following mill:

- Curve sawing is available and cant guiding is therefore selected as “half taper” curve sawing, meaning that the saws will follow the centre-line of the cant;
- Re-sawing is not considered and selected as “Off”;
- Log and dealsaw kerf is selected as 4 mm;
- Wane was not allowed and thus set to 0%;

- The edger kerf was selected as 5 mm;
- The board sizes were defined as that typically used in the South African context:

Table 5: Wet and dry board sizes in the South African context

	Wet (mm)	Dry (mm)
Thickness	27	25
	41	38
Width	81	76
	120	114
	160	152
	240	228

The sawing patterns for logs of the SED sizes as mentioned are illustrated in figure 11 below.

For this study, which focuses on log positioning during breakdown, the simulations were done for the following range of positions (see figure 7 for positioning variables):

- Rotation of 0 to 345 degrees at an increment of 15 degrees, where 0 degrees should be interpreted as the horns up position (24 rotational positions).
- Offset of -30 mm to 30 mm at an increment of 5 mm (13 offset positions).
- Skewing of -30 mm to 30 mm at an increment of 5 mm (13 skewing positions).

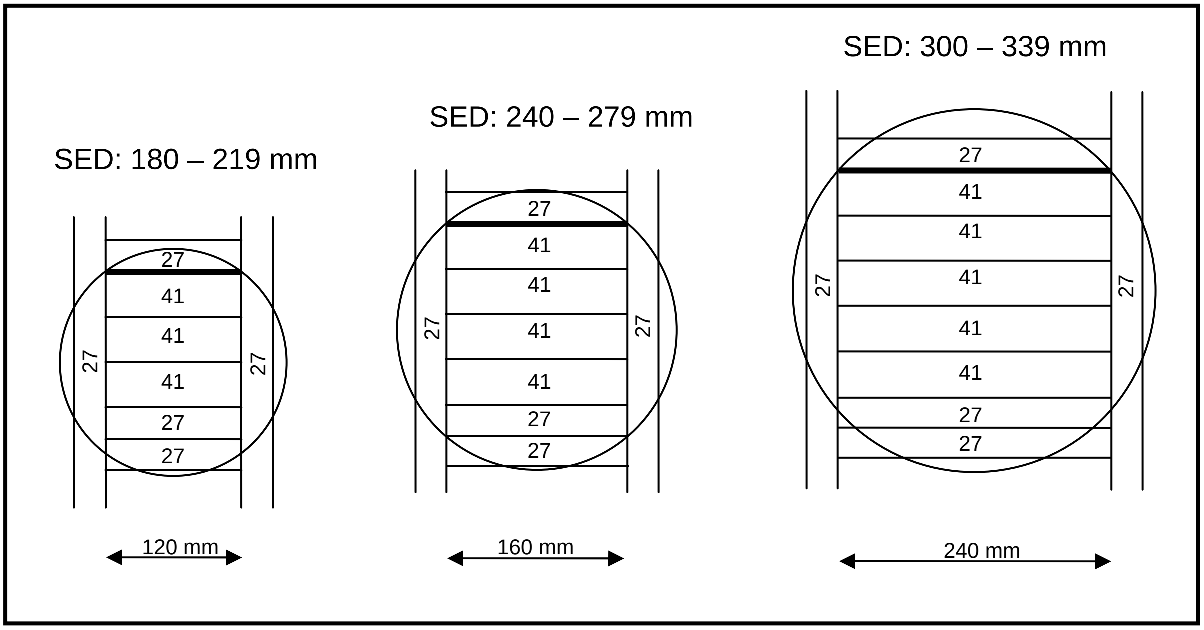


Figure 11: Wet size sawing patterns used in the simulation.

The following annotation for the positioning variables is used: [rotation; offset; skewing].

For the effective evaluation of heuristic methods an exhaustive search of all combinations of the positioning variables was firstly needed. The simulation software was therefore used to attain recovery values for all 4056 positioning combinations (24 x 13 x 13) for each of the 60 logs. A total of 243360 simulations were therefore run on the software.

4.2 The simulated annealing algorithm

For the present study the results of the simulation software were imported into Microsoft Excel where the data could be structured and analysed. The logic and algorithm of the SA meta-heuristics was programmed into the Excel sheets. See Appendix D for the layout of the program. It was done in a number of steps, as follows:

- Firstly each unique combination was given a six digit code. The code consisted of three pairs of two digit numbers where each pair indicated the incremental step for each variable. For each pair the number 10 was given to the minimum value in the range. This was done to simplify the lookup of data. The code could thus be directly translated to the specific rotation, skewness and offset of the specific log position. For example the [0, 0, 0] position was numbered as 101616. See Appendix D for a sample of the unique codes.

- Secondly a neighbourhood was defined for each combination. 4 incremental steps in either direction of each variable for the combination under consideration were selected as the neighbourhood. The shift for each variable was selected by a random number between 0 and 1.
- The starting position for the algorithm was selected as [0, 0, 0]. The recovery value for this position was attained from the simulation data.
- By selecting 3 random numbers between 0 and 1 the next combination in the current neighbourhood could be selected. The new values were capped to stay within the range of available combinations. In other words when a neighbour with an offset of more than 30 mm was suggested by the neighbourhood function, the offset was capped to 30 mm. In the case of rotation when a rotation of minus 30 degrees was suggested, the rotation was corrected to 330 degrees.
- A new combination in the neighbourhood of the existing combination under consideration was therefore selected and the recovery value was attained from the simulation data.
- Whether the focus will shift to this new combination will depend on two logical tests.
- The first logical test is whether the new combination's recovery is greater than the existing one.
- The second logical test is defined for the Simulated Annealing algorithm in paragraph 3.5.1. For this test a temperature value with a specific cooling regime needs to be defined. Various options for these user defined operators were evaluated.
- If either of the above logical tests is true, the focus will shift to the new combination.
- The process was repeated until a specified number of unique combinations were evaluated.

- The results for each log was recorded and presented for evaluation.

The Macro programming function was used to simplify the process of running the programmed meta-heuristic on each successive log. For instance, with the amount of data and calculations needed in each Excel workbook, the computer that was used ran out of memory on some occasions. A macro was utilized to run the heuristic method on one sheet at a time and to then delete the calculations after the calculated values were attained. Successive sheets or logs were then investigated by the macro.

Another meta-heuristic method was developed during the course of this study. This meta-heuristic will now be explained in the following paragraph. For convenience it will be called the Tentacle algorithm.

4.3 The development of the Tentacle algorithm

Heuristic methods are designed to not get stuck in a local optimum. With Simulated Annealing, for example, the focus will shift to weaker combinations when a stochastic operator is smaller than a diminishing temperature value. This allows a greater area in the search space to be evaluated by not restricting the focus to only shift to better combinations. The tentacle algorithm was conceived after noting that allowing the focus to shift to weaker combinations, as in SA, was not always productive. As mentioned earlier, experience has shown that the centered horns-up position in log volume recovery yields good results and a near optimal combination can be expected near this position. This position is therefore the logical starting point in a heuristic search of an 'optimum' volume recovery. To shift the focus to weaker combinations (to allow a greater area in the search space to be evaluated) could in this case take the focus further away from the expected 'optimal' recovery; making it harder for the heuristic to find its way back to good combinations. **Note:** The concept of a search tentacle using hill climbing is fairly simple and it may well be used in other algorithms as well. This algorithm has been developed independently from any other method and it's uniqueness is the specific application area and method of creating the tentacle path.

The Tentacle algorithm that was suggested is based on a classic hill-climbing algorithm where all the combinations directly neighbouring the combination under consideration are evaluated. As soon as an improved combination is found the focus will jump to this combination. The process will then be repeated until no direct neighbour renders better results. In other words the focus has come to the top of a hill. Now, instead of accepting weaker combinations to escape this potentially poor local optimum the algorithm will then start looking for combinations further away than the direct neighbours of the combination in focus. The analogy would be of a hill climber that has tentacles, which it could extend to feel the surrounding space through the dense fog for elevations higher than the current one without stepping down to lower elevations. The tentacles can then be extended to feel further and further into the search space. The neighbourhood will thus grow in steps until a combination is reached that gives improved results. The algorithm will however not do an exhaustive search of this increasing neighbourhood. It will probe this selection of combinations almost like the tentacles of an octopus growing out of the focus point. The tentacles will then keep growing with the increased neighbourhood until an improved combination is found. If an improved combination is found the focus will then jump to this stronger combination and the tentacles will shrink so that they can only reach the direct neighbours and the algorithm will go back into hill-climbing mode. The parameters that were altered to optimise and test this algorithm on the Simsaw5R results was: (a) Tentacle growth rate and (b) the use of flexible tentacles where random combinations in reach of the tentacles are tested instead of only the ones on a straight line extending from the focal point. The number of tentacles was kept constant.

The structure of the Tentacle algorithm can be described as follows:

Tentacle algorithm:

procedure tentacle algorithm

begin

$t \leftarrow 1$

Initialization

Select the [0, 0, 0] starting position for the focus of the algorithm as $x^{now} \in X$, where X is the search space

Record the current best known position as $x^{best} \leftarrow x^{now}$
And the best evaluation as $eval(x^{best}) \leftarrow eval(x^{now})$

repeat

repeat

select a new point $x^{next} \in N(x^{now}, t)$ where $N(\,)$ is the neighbourhood function of x^{now} . The variable t is defined as the focus shift. It defines the number of steps the focus shifts in either direction of the current focal point for any variable of the combinatorial optimisation problem. Remember the search space is discreet and in layman's terms each combination is a number of steps per variable away from another. $N(\,)$ is the set of all the possible combinations with these steps for each variable around x^{now}. An example follows below to illustrate this neighbourhood.

if $eval(x^{next}) > eval(x^{best})$

then $x^{best} \leftarrow x^{next}$; $x^{now} \leftarrow x^{next}$

and $eval(x^{best}) \leftarrow eval(x^{next})$

until $N(x^{now}, t)$ is exhausted

$t \leftarrow t + 1$

until (halting criterion)

This neighbourhood will now be further explained on the hand of an example in two dimensions, i.e. a combinatorial problem with only two variables.

Assume a two dimensional combinatorial optimisation problem. See figure 12 below. The starting position, is selected as E5. With t=1 the neighbourhood is all combinations reachable with a focal shift of 1 in either direction per variable. Thus D4, E4, F4, D5, F5, F6, E6, and D6 constitute this neighbourhood. When t increases to 2 the neighbourhood will change to C3, E3, G3, G5, G7, E7, C7, and C5. When t increases to 5 the neighbourhood will change to A1, E1, I1, I5, I9, E9, A9 and A5.

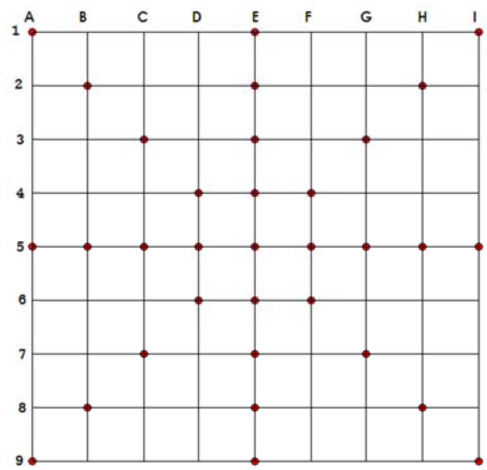


Figure 12: Diagram showing grid of possible combinations A1 to I9

4.3.1 Parameter changes considered with Tentacle algorithm

As with Simulated Annealing the Tentacle algorithm was also tested with different sets of parameters. The initial algorithm let the tentacles grow in steps of 1, i.e. firstly the direct neighbours are considered, if no better combination is found the neighbour directly following the first is selected and so on. The tentacles also only grew in straight lines or in other words linearly from the focal point. These two features were altered to see if the performance of the meta-heuristic might improve.

A feature called accelerated growth was implemented and tested. With **accelerated growth** the tentacles were programmed to have an exponential growth rate, i.e. after failing to attain a better combination amongst the direct neighbours the tentacles would skip the following one, and then without a better combination the following two. In terms of the algorithm, when no better combinations were encountered in a search in a neighbourhood, the term t would be changed to t + 2 and if, again, no better combinations were encountered, to t + 5 and so on.

A second feature was also implemented and tested which was called **flexible tentacles**. With flexible tentacles, the tentacles were thought of as flexible, allowing any combination to be tested within a specific distance of a particular tentacle. In other words, the tentacles do not need to extend in a linear fashion away from the tentacle. This might sound more intricate than it really is. It can be briefly explained as follows, a more detailed explanation will follow in the next paragraph. Assume the algorithm in the first instance (with rigid tentacles) has reached a local optimum of a two dimensional combinatorial problem as illustrated in figure 13 below. The focal point now is combination E5. Further assume the algorithm has extended the tentacle length to 4 without finding a better combination. Looking at figure 13 below, the tentacle pointing in the one o'clock thirty minutes direction or in the North Easterly direction will be pointing to point I1 on the grid. Now, instead of the tentacles only moving in a linear direction away from the focal point which makes combination I1 the combination to be tested in this instance assume the area indicated by the grey block in figure 13 is in reach of the tentacle and that a randomly selected combination within this selection is considered.

It should be evident that the particular tentacle discussed above might overlap with the tentacles at 12 o'clock and 3 o'clock (Northerly and Easterly direction) and that these tentacles have become redundant with this particular set up.

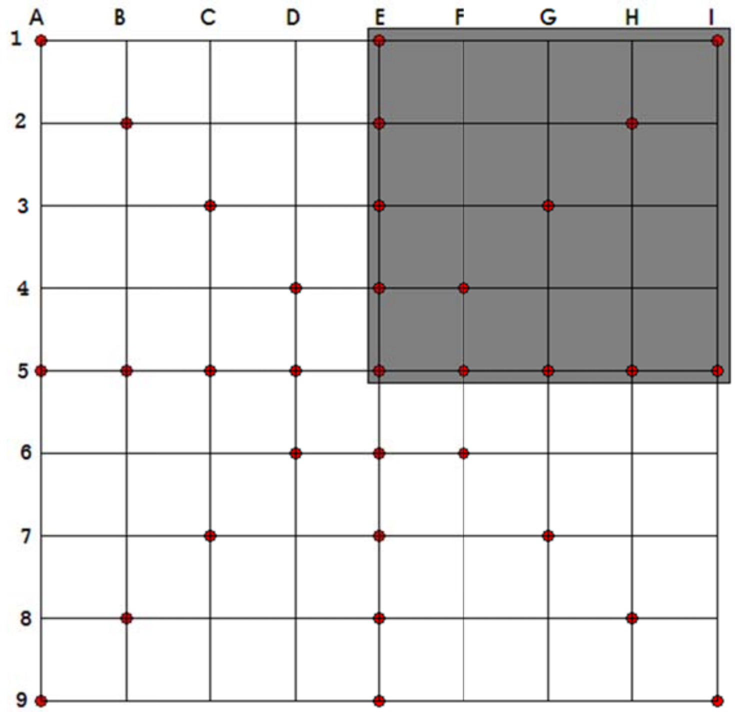


Figure 13: The neighbourhood of one flexible tentacle.

4.3.2 Definition of the Neighbourhood of the Tentacle algorithm

It should be clear by now that the Tentacle Meta-heuristic has two important characteristics. The first is that of a classic hill climbing algorithm. In other words the algorithm evaluates combinations directly adjacent to the focal point and set the focus to combinations that yield improved results. As soon as the algorithm reaches a local optimum the next characteristic becomes apparent. This is that the algorithm will probe the combinations close to the attained local optimum but not the ones directly next to the local optimum as they have already been evaluated during the hill climbing stage. These combinations close to the local optimum is referred to as the neighbourhood. This neighbourhood grows with a prescribed pattern. This growing neighbourhood, or in other words the combinations in reach of the tentacles, will be explained in 4 steps:

- Firstly, where the direct neighbours will be defined for a two variable problem;

- Then, where the neighbourhood for the same problem has grown for two cycles;
- Then, where the neighbourhood has grown for another cycle and where 'flexibility' of the tentacles is introduced;
- Lastly the ease with which the algorithm can be used for higher order problems is illustrated with the definition of the neighbourhood for a 5 variable problem.

Step 1: From the above flexible tentacle explanation it is evident that when this heuristic method probes the neighbourhood for possibly stronger combinations the heuristic is instructed to evaluate points which are a number of steps away from the current focal point. For instance from figure 12 when E5 is the current focal point, D5 would be one step to the left, point D4 would be one step to the left and one step up, etc. This is not necessarily useful with combinatorial problems where the variables are dependent on each other as per example with the travelling salesman problem the sequence of cities is important and each sequence will be allocated to a distinct city.

(It is also important to remember that A to I and 1 to 9 are ranges of variables in a two dimensional combinatorial optimisation problem and each node is a specific combination of these variables. A two variable combinatorial problem would most likely be too simple for the use of meta-heuristics, but this example is used as explanation of how this meta-heuristic would move through the search space)

It is possible, therefore, to define the immediate neighbours of the focal point in the following illustration (Figure 14). The numbers -1, 0 and 1 is meant as a move of that variable by 1 step in the negative direction, no move or 1 move in the positive direction, respectively. The 8 direct neighbours of node E5 is illustrated in the grid next to the table. Note that the size of the neighbourhood with (n) variables is:

Neighbourhood size = $3^n - 1$.

Step 1: Defining direct neighbours for tentacle algorithm

Neighbour number	Steps from focal point for Variable 1 (1 to 9)	Steps from focal point for Variable 2 (A to I)	Direct neighbours for node E5
1	1	0	E6
2	1	1	F6
3	0	1	F5
4	-1	1	F4
5	-1	0	E4
6	-1	-1	D4
7	0	-1	D5
8	1	-1	D6

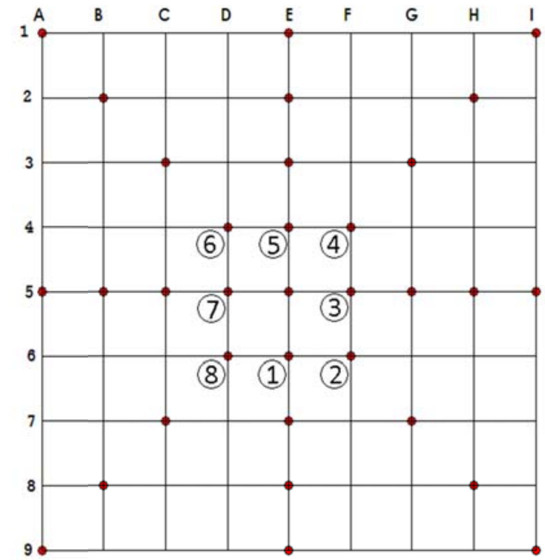


Fig 14: Defining the direct neighbours as steps away from focal point (Step 1).

Step 2: To illustrate the ease with which the tentacle algorithm incorporates tentacle growth we introduce a constant, namely Tentacle Length (TL). This length will increase as per the definition in the algorithm. Let us assume the length increases with 1 unit each cycle of not finding a better combination. Further assume the algorithm has completed two cycles of increasing the neighbourhood without finding a better combination, the tentacle length is thus 3 units. The neighbourhood of node E5 will then be defined in the next illustration (Figure 15).

Step 2: Defining neighbours of node E5 with Tentacle Length of 3, thus TL = 3

Neighbour number	Steps from focal point for Variable 1 (1 to 9)	Steps from focal point for Variable 2 (A to I)	Combination under evaluation for node E5 with Tentacle Length (TL)= 3
1	1 . TL	0 . TL	E8
2	1 .TL	1 . TL	H8
3	0 . TL	1 . TL	H5
4	-1 . TL	1 .TL	H2
5	-1 . TL	0 . TL	E2
6	-1 . TL	-1 . TL	B2
7	0 . TL	-1 . TL	B5
8	1 .TL	-1 . TL	B8

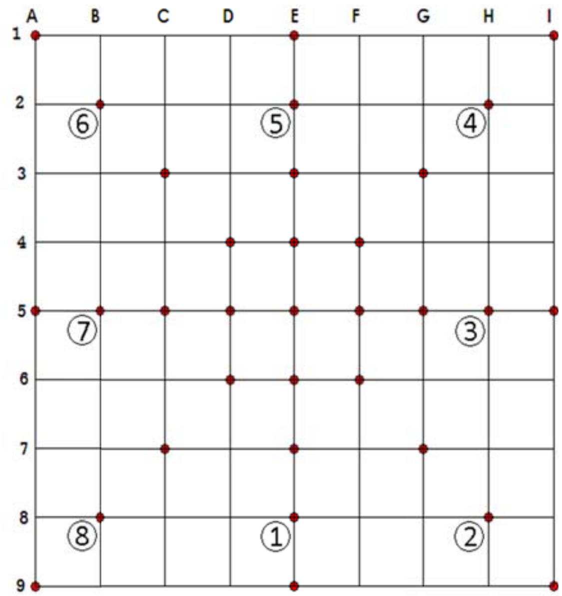


Fig 15: Defining the neighbourhood of the tentacle algorithm as steps away from the focal point and with tentacles the length of 3 units (Step 2).

Step 3: With the definition of the neighbourhood explained as steps away from the focal point and with the concept of Tentacle Length added we can now explain the simplicity of adding flexibility to the tentacles. We do this by multiplying each step away from the focal point by a random number between and including zero and one. This product is then rounded to the nearest integer as fractions of a step are obviously not possible. Instead of having a set of distinct combinations as the neighbourhood, we will have a set of ranges within which one random combination will be evaluated. As mentioned earlier the introduction of these random numbers will cause the ranges making up the neighbourhood to overlap and the ranges can be reduced to 4 instead of 8. See figure 16 below for a table and illustration of this concept.

Step 3: Defining neighbours of node E5 with Flexible Tentacles of Tentacle Length (TL) 4

Neighbour number	Steps from focal point for Variable 1 (1 to 9)	Steps from focal point for Variable 2 (A to I)	Combination under evaluation for focal point on node E5 with Tentacle length (TL)= 4 multiplied by random number from 0 to 1 and rounded to nearest integer
2	1 .TL . Rand[0,1]	1 . TL . Rand[0,1]	Random node from block 1, including boundary
4	-1 . TL . Rand[0,1]	1 .TL . Rand[0,1]	Random node from block 2, including boundary
6	-1 . TL . Rand[0,1]	-1 . TL . Rand[0,1]	Random node from block 3, including boundary
8	1 .TL . Rand[0,1]	-1 . TL . Rand[0,1]	Random node from block 4, including boundary

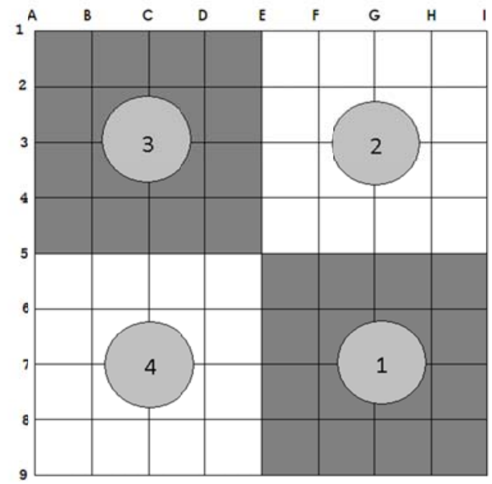


Fig 16: Defining the neighbourhood of the tentacle algorithm as steps away from the focal point and with flexible tentacles the length of 4 units (Step 3).

Step 4: Lastly, as per example the neighbourhood of a 5 variable combinatorial problem will be defined with the help of table 6. Each neighbour is once again defined as a number of steps away from the focal point. The function f(TL) referred to in the table is TL.Random [0,1] rounded to the nearest integer. It should be noted that the neighbourhood size is a function of the number of variables (n) such that:

Neighbourhood size = 2^n

It should be mentioned that the table has all the combinations of a binary number but with 0 replaced by -1.

Table 6: Definition of neighbourhood as steps away from focal point for 5 variable combinatorial optimisation problem f(TL) = (TL . Random[0,1])rounded to the nearest integer.

Neighbour number	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5
1	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)
2	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)
3	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)
4	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)
5	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)
6	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)
7	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)
8	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)
9	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)
10	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)
11	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)
12	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)
13	-1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)
14	-1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)
15	-1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)
16	-1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)
17	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)
18	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)
19	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)
20	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)
21	1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)
22	1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)
23	1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)
24	1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)
25	1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	-1 . f(TL)
26	1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)	1 . f(TL)
27	1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)	-1 . f(TL)
28	1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)	1 . f(TL)
29	1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)	-1 . f(TL)
30	1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)	1 . f(TL)
31	1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)	-1 . f(TL)
32	1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)	1 . f(TL)

4.4 Two-dimensional examples of the Tentacle algorithm

As an example two hypothetical log positioning searches will be described below.

Table 7 illustrates a hypothetical tentacle search process in a two-dimensional space with rotation and offset as the only two variables. The numbers in each block indicate the volume recovery obtained for a specific combination and the superscripts indicate the search sequence. The grey-coloured blocks indicate the combinations that are focal points during a specific stage of the search. In the first iteration the 0º and 0 mm position is the focal point and this position yields a volume recovery of 47%. In the second iteration the [15º, 0 mm] combination is evaluated and yields a volume recovery of 43%. Since this is smaller than the current focal point, the next neighbor [15º, 5 mm] is evaluated, which yields a 45% recovery. In the same way all the direct neighbors are evaluated and – in this example - yield inferior results to the current focal point. In the next phase the second order, third order and fourth order neighbors are evaluated in iterations 10 to 32. Note the linear outward move of the search. At the 32nd iteration in position [0º, -20 mm] the volume recovery of 48% is superior to the current focal point and subsequently this position is selected as the new focal point. The search process around the new focal point is continued in the same way until a new focal point is found in iteration 44 at the position [30º, -30 mm].

Table 7. An example of a two-dimensional tentacle search process with linear tentacles. The numbers in each block represent the volume recovery at a specific log position and the superscripts are the iteration numbers in the search. Grey colored blocks represent focal points.

		Offset (mm)											
		-30	**-25**	**-20**	**-15**	**-10**	**-5**	**+0**	**+5**	**+10**	**+15**	**+20**	**+25**
Rotation (degrees)	**75º**												
	60º							44^{26}				42^{27}	
	45º				45^{25}			45^{18}			43^{19}		
	30º	**49^{44}**		46^{40}		44^{17}		43^{10}		44^{11}			
	15º		47^{39}	47^{33}	46^{34}		43^{9}	43^{2}	45^{3}				
	0º	46^{43}	47^{38}	**48^{32}**	46^{24}	45^{16}	45^{8}	**47^{1}**	42^{4}	41^{12}	40^{20}	40^{28}	
	345º		45^{37}	47^{36}	46^{35}		44^{7}	44^{6}	43^{5}				
	330º	45^{42}		46^{41}		45^{15}		43^{14}		38^{13}			
	315º				45^{23}			43^{22}			35^{21}		
	300º			46^{31}				42^{30}				32^{29}	
	285º												

A modification of the tentacle algorithm, similar as described above, has also been tested. The modified version of the algorithm does not follow a fixed linear path away from the focal points, but uses a random process to select a new position within a fixed range away from the focal point. This version could be described as having flexible tentacles. In Table 8 the search process where the modified tentacle algorithm is used, is illustrated. Each tentacle moves outwards within the range indicated by the bold lines. Instead of selecting a pre-determined position with each iteration, any position within a fixed range can be selected in a random fashion. For example, at iteration 26 in the example shown in Table 8, the position 60º and -5 mm was selected randomly from the options [60º, -10 mm], [60º, -5 mm], [60º, 0 mm], and [60º, +5 mm].

Table 8. An example of a two-dimensional tentacle search process with flexible tentacles. The bold lines indicate the borders within which each tentacle is allowed to search. The numbers in each block represent the volume recovery at a specific log position and the superscripts are the iteration numbers in the search. Grey colored blocks represent focal points

		Offset (mm)											
		-30	**-25**	**-20**	**-15**	**-10**	**-5**	**+0**	**+5**	**+10**	**+15**	**+20**	**+25**
Rotation (degrees)	**75º**					42^{34}							42^{35}
	60º						43^{26}				43^{27}		
	45º			46^{33}	45^{25}			42^{18}		43^{19}			
	30º					44^{17}		43^{10}	43^{11}				
	15º				45^{24}		45^{9}	45^{2}	45^{3}			41^{28}	
	0º	48^{47}	48^{41}	47^{32}		45^{16}	44^{8}	**48^{1}**	43^{4}	43^{12}	40^{20}		
	345º	47^{46}	**49^{40}**	46^{42}			47^{7}	44^{6}	45^{5}	39^{13}			
	330º	46^{45}	47^{44}	46^{43}		46^{15}		44^{14}					40^{36}
	315º			45^{31}		45^{23}	43^{22}			35^{21}			
	300º							42^{30}			33^{29}		
	285º		45^{39}						41^{38}				32^{37}

4.5 Comparison of different search strategies

Two meta-heuristics were evaluated in this study. With all the combinations simulated the global optimum in the search space was known. The results obtained by the meta-heuristics were evaluated against the global optimum after 50, 90 and 350 iterations.

Two additional meta-heuristics were evaluated in other studies but with the same simulated data. These were the Population Based Incremental Learning method and the Particle Swarm Optimisation method

(Wessels, 2008 and Smit, 2009). The results of all the methods were compared with the strategy of exhaustively searching smaller high quality regions of the search space.

For comparison purposes the method of randomly selecting positions within the search space was also considered. For each log 50, 90 and 350 positions were selected randomly from the positioning range of 4056 positions. The best performing position of the 50, 90 and 350 randomly selected positions respectively of each log were then used to obtain an average performance of this strategy in terms of volume recovery.

5. Results and Discussion

Results of three other studies are used within this research work and some of these results will also be reported here for discussion and comparative purposes.

5.1 Simulation results

Results of this study are also reported by Wessels (2009b). The sawing of each log was simulated in 4056 different positions and the optimal position in terms of volume recovery was recorded for each log. The optimal log positions for these logs are depicted in the graphs below (see figure 17).

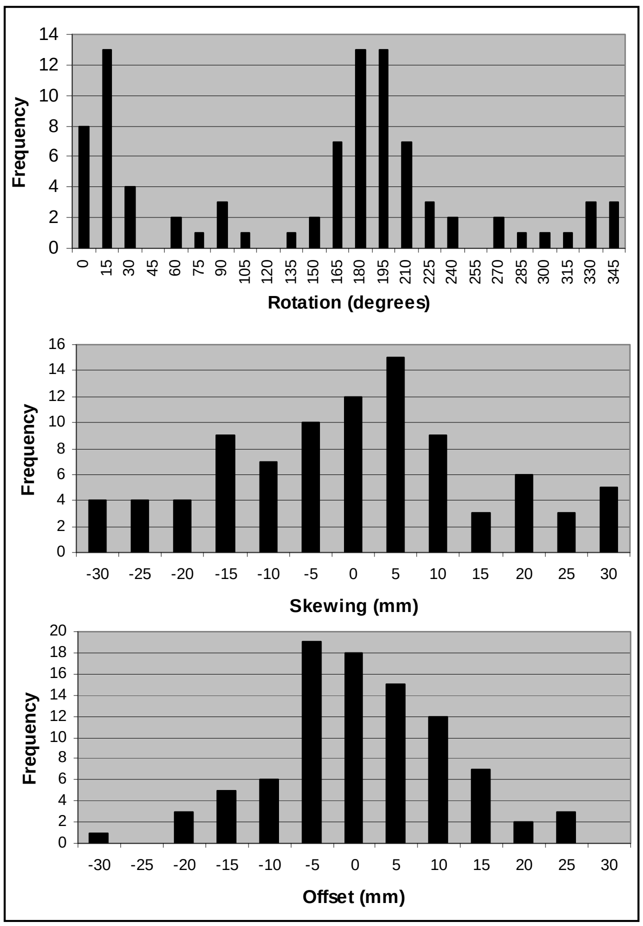


Fig. 17 Histogram of optimal position values (adapted from Wessels 2009b)

From the results in Figure 17 it is clear that the distribution of optimal positioning results confirms that there is a higher frequency of optimal positions around the conventional log positions used (i.e. “horns-up”, “horns-down”, centered, “full-taper”), but that in most cases the optimal log position is different to the exact conventional position.

With log volume recovery, experience has shown that the centred horns-up or horns-down position yield good results. The histograms in Figure 17 show that optimal positions often occur close to this position. By reducing the range the number of combinations is reduced. The reduced search space could potentially be exhaustively searched as an alternative to using heuristics. The effect of reducing the search ranges to that around the high quality areas in the search space is depicted in the graph below (see Figure 18).

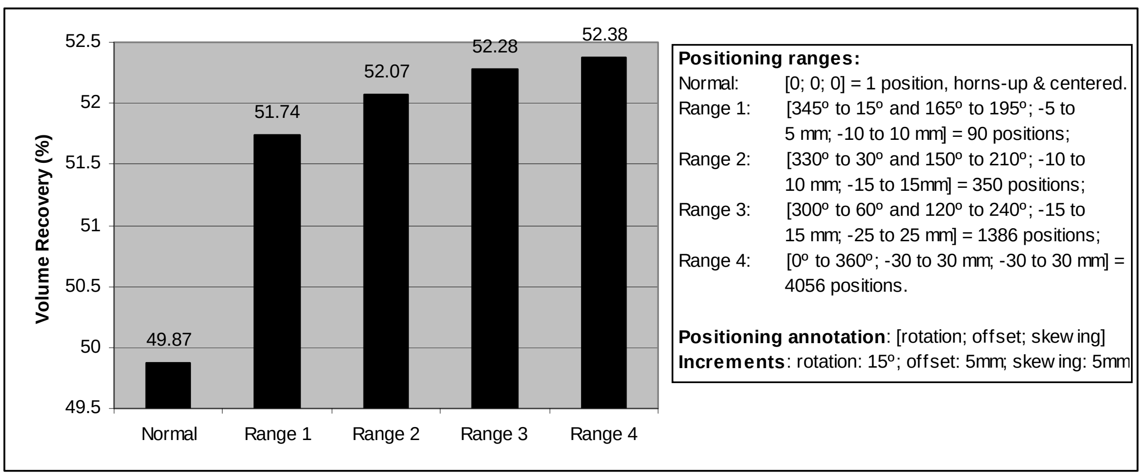


Figure 18. The average maximum volume recoveries obtained within four positioning ranges for the 60 sample logs (Wessels, 2009b).

The difference between the average maximum volume recovery in the largest range (4056 combinations) and smallest range (90 combinations) is relatively small at 0.64%. This illustrates the fact that a significant gain in volume recovery can be achieved even if only a relatively small range of positioning combinations is searched around the centered, “horns-up” and “horns-down” positions. The average maximum volume recovery in the smallest range of 90 combinations is still 1.87% better than the normal centered and “horns-up” position.

5.2 Comparison of search algorithms

The results, i.e. the recovery obtained after executing a set number of iterations (50, 90 and 350 in this instance) for Simulated Annealing and Tentacle algorithm, with three settings of its parameters can be found in Appendix B. The maximum recovery in the search space is also shown together with the recovery which would result from the centred horns up position. A strategy of randomly selecting positioning solutions from the search range is also shown for comparative purposes. The results of all these search strategies, including PBIL (Wessels 2008) and PSO (Smit, 2009), are summarised below.

Table 9: Comparison between different search algorithms to find optimal log positioning coordinates for 60 pine logs. Dark gray shaded cells indicate the best performing algorithm and light grey shaded ones indicate the second-best performing algorithm for a specific number of simulations. Included are results from an exhaustive search in smaller ranges, random selections in the search space, as well as the global optimum result in the search space.

	Parameters				Average maximum volume recovery		
				Line	50*	90*	350*
Population based incremental learning (PBIL)	Learning rate	Mutation probability	Mutation shift				
	0.1	0.03	0.05	1	50.73 %	51.20 %	51.78 %
	0.1	0.1	0.1	2	50.80 %	51.20 %	51.85 %
	0.2	0.03	0.05	3	50.95 %	51.33 %	51.57 %
	0.2	0.1	0.1	4	50.89 %	51.31 %	51.71 %
Particle Swarm Optimisation (PSO)	Velocity = 0.5			5	50.82 %	51.06 %	51.06 %
	Velocity = 1.0			6	51.31 %	51.43 %	51.45 %
	Velocity = 1.5			7	51.55 %	51.61 %	51.61 %
Simulated annealing (SA)	Starting temp = 5 000; 10% cooling			8	50.88%	51.24%	51.82%
	Starting temp = 50 000; 10% cooling			9	50.80%	51.13%	51.81%
	Starting temp = 5 000; Linear cooling			10	50.78%	51.15%	51.80%
Tentacle	Rigid tentacles; constant growth			11	51.50%	51.62%	51.90%
	Flexible tentacles; constant growth			12	51.52%	51.70%	51.89%
	Rigid tentacles; accelerating growth			13	51.28%	51.43%	51.57%
Random selection of positions within search space				14	50.86%	51.24%	51.79 %
Exhaustive search	Range 1 (see Figure 18)			15		51.74 %	
	Range 2 (see Figure 18)			16			52.07 %
Global optimum in search space (4056 simulations)				17	52.38 %		
Recovery at centered, "horns-up" position (1 simulation)				18	49.87 %		

From the table above it can be seen that after 50 iterations the PSO meta-heuristic has reached an average maximum recovery of 51.55% for the log sample used. A velocity of 1.5 was used in this instance. This was the best performing algorithm after 50 iterations. The second best was the Tentacle algorithm with flexible tentacles with a value of 51.52%. With a global maximum average recovery of 52.38% and a standard "horns-up" recovery of 49.87% it is evident that 67% of the potential increase in recovery from an exhaustive search is attained after 50 iterations using the PSO algorithm with a velocity of 1.5. The worst performing meta-heuristic after 50 iterations was the PBIL with the first set of parameters. This meta-heuristic attained a value of 50.73%. By selecting 50 random combinations from the search space an average maximum recovery of 50.86% was attained, indicating that the PBIL is very inefficient after only 50 iterations.

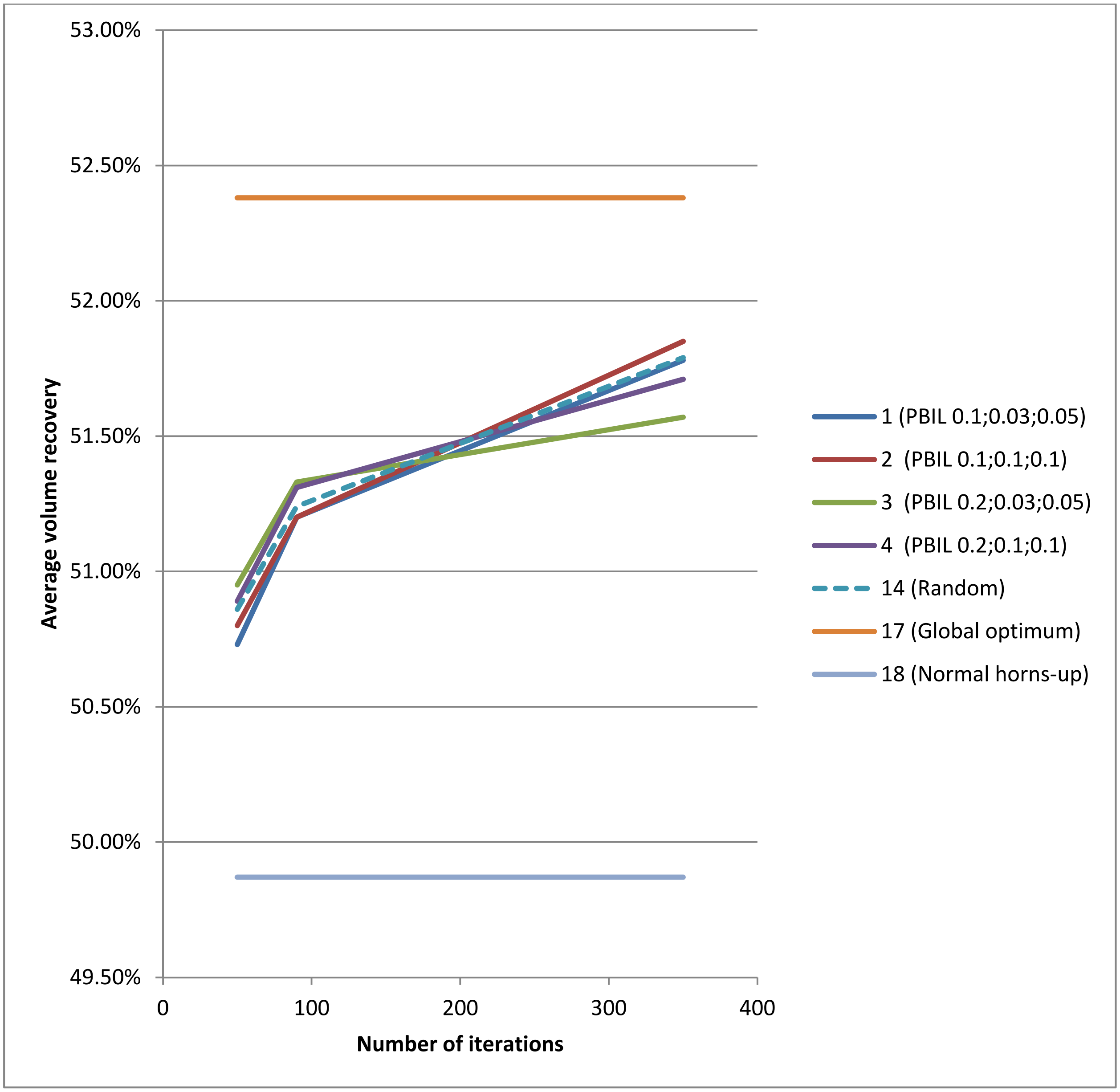
After 90 iterations the Tentacle algorithm with flexible tentacles attained an average maximum recovery of 51.70%, which was the best value attained by the meta-heuristics. The second best performance was attained by the Tentacle algorithm with rigid tentacles and constant growth. The tentacle algorithm attained 73% of the potential increase in recovery after 90 iterations. The exhaustive searching of a smaller range, containing 90 combinations in the high quality area, yielded a 51.74% average maximum recovery, thus outperforming all of the meta-heuristics. After 90 iterations the worst performing meta-heuristic was the PSO meta-heuristic with a velocity of 0.5. The value obtained by this meta-heuristic was 51.06%, which is lower that the value obtained by selecting 90 random combinations from each search space. The random selection yielded a value of 51.24%.

After 350 iterations the best performing meta-heuristic was the Tentacle algorithm with rigid tentacles, attaining a value of 51.90%. The second best performance was from the Tentacle algorithm with flexible tentacles. The tentacle algorithm attained 81% of the potential improvement of an exhaustive search of the search space after evaluating 350 combinations. The exhaustive searching of a reduced range of 350 combinations in the high quality area again outperformed all the meta-heuristic by yielding a result of 52.07%. The worst performing meta-heuristic after 350 iterations was the PSO with a velocity of 0.5. It reached a value of 51.06%, which again is lower than selecting 350 random combinations from each search space. The random selections yielded a value of 51.79%.

The results of the random selection of 50, 90 or 350 iterations are surprising, considering the fact that this method will select solutions throughout the full search space. From table 9 it can be seen that some of the more complex meta-heuristics were actually outperformed by random selections. These algorithms must have directed their search, in most cases, to below-average areas of the search space. Using random selections has the advantage of having no prejudice for the selection of a particular combination. Therefore there is no risk of the algorithm getting stuck in a local optimum. On the other hand, however, there is also no learning of where good results can be found. The four meta-heuristics used are now individually evaluated against the selection of random values, which is designated as benchmark in this report. The normal and exhaustively searched average recoveries are also shown on each graph.

5.2.1 PBIL

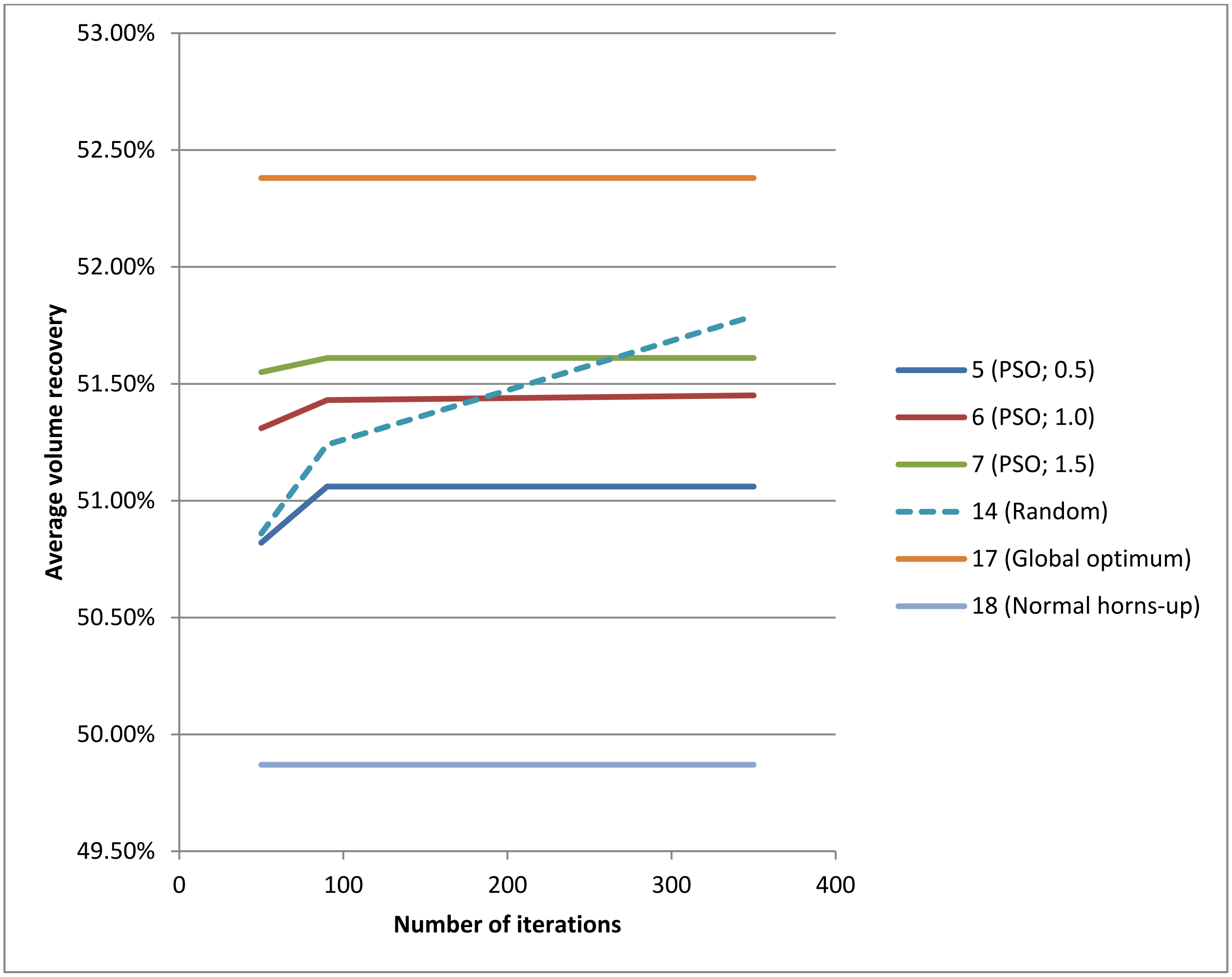
See graph 1 below for a graphic presentation of the PBIL results attained after 50, 90 and 350 iterations. Each of the four variations is presented with the legend referring to the line number from table 9. The broken line represents the results attained by the random selection of combinations.



Graph 1: Graphic presentation of PBIL results

From the results it can be seen that the performance characteristics change with different user parameters. For the PBIL meta-heuristic for example a learning rate of 0.2 instead of 0.1 increased the initial efficiency of the meta-heuristic but decreased the effectiveness after 350 iterations. Similarly, the probability and size of a mutation shift affects the efficiency. In the case where a learning rate of 0.1 was used, the mutation probability and mutation shift of 0.1 rendered better results. The best result after 350 iterations was attained from the PBIL with a learning rate of 0.1; a mutation shift of 0.1 and a mutation probability of 0.1. In this instance 78.9% of the potential increase in recovery was achieved after 350 iterations where the random selection of values reached 76.5%. In this instance the results after 50 iterations were poorer than the random selection of values, illustrating the desirability of a greater number of iterations in this case.

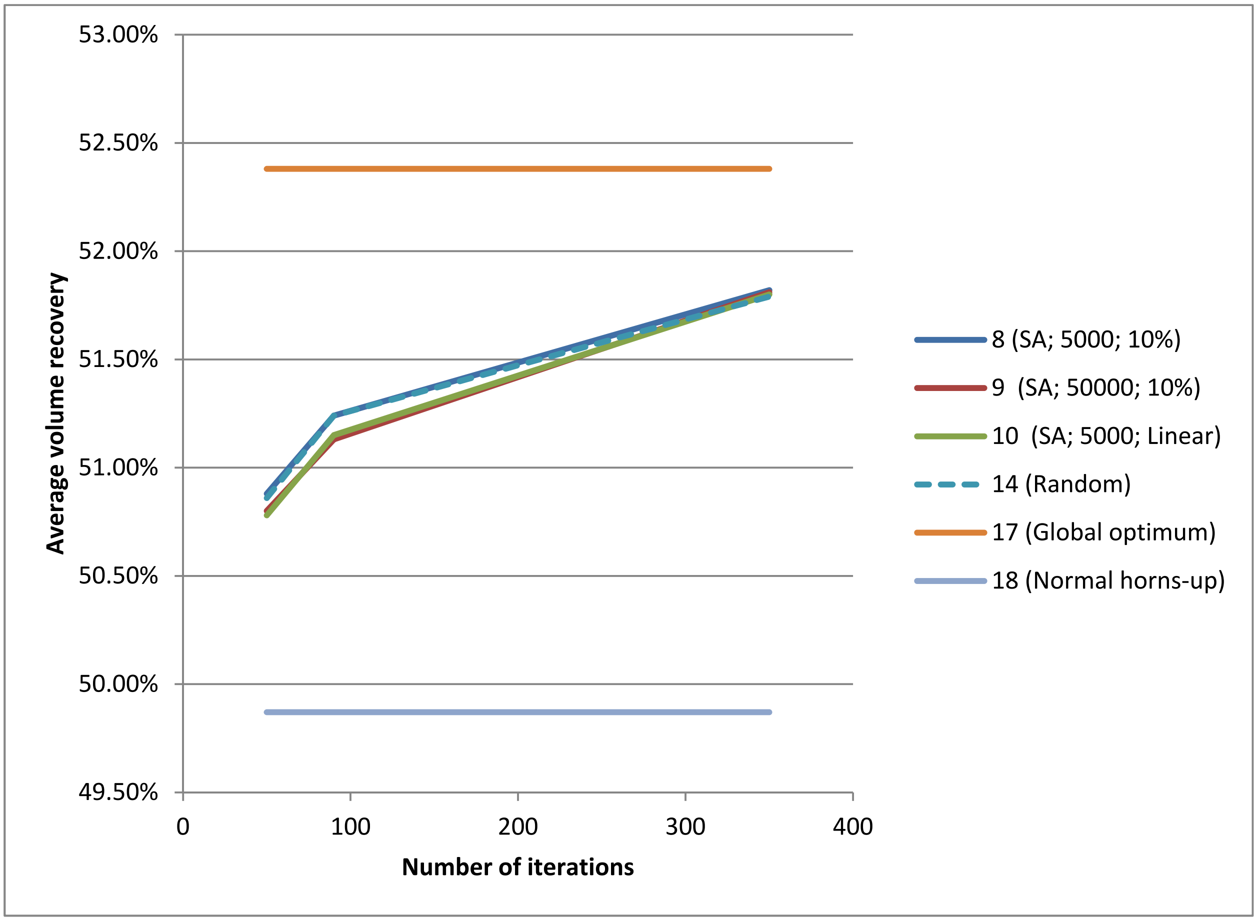
5.2.2 PSO



Graph 2: Graphic presentation of PSO results

With the PSO only the maximum speed of the particles was used as user defined parameter. From the results it can be seen that the results did not improve much after 90 iterations despite relatively good results after 50 iterations. The PSO meta-heuristic with a velocity of 1.5 reached an average recovery of 51.55% after 50 iterations, which makes it the best performing meta-heuristic under consideration at that point. The meta-heuristic reaches convergence at around 90 iterations, however, as seen in graph 2. This meta-heuristic, with the parameters used, converged fairly quickly. It should, therefore, only be used when less than 90 iterations are possible during the log positioning process.

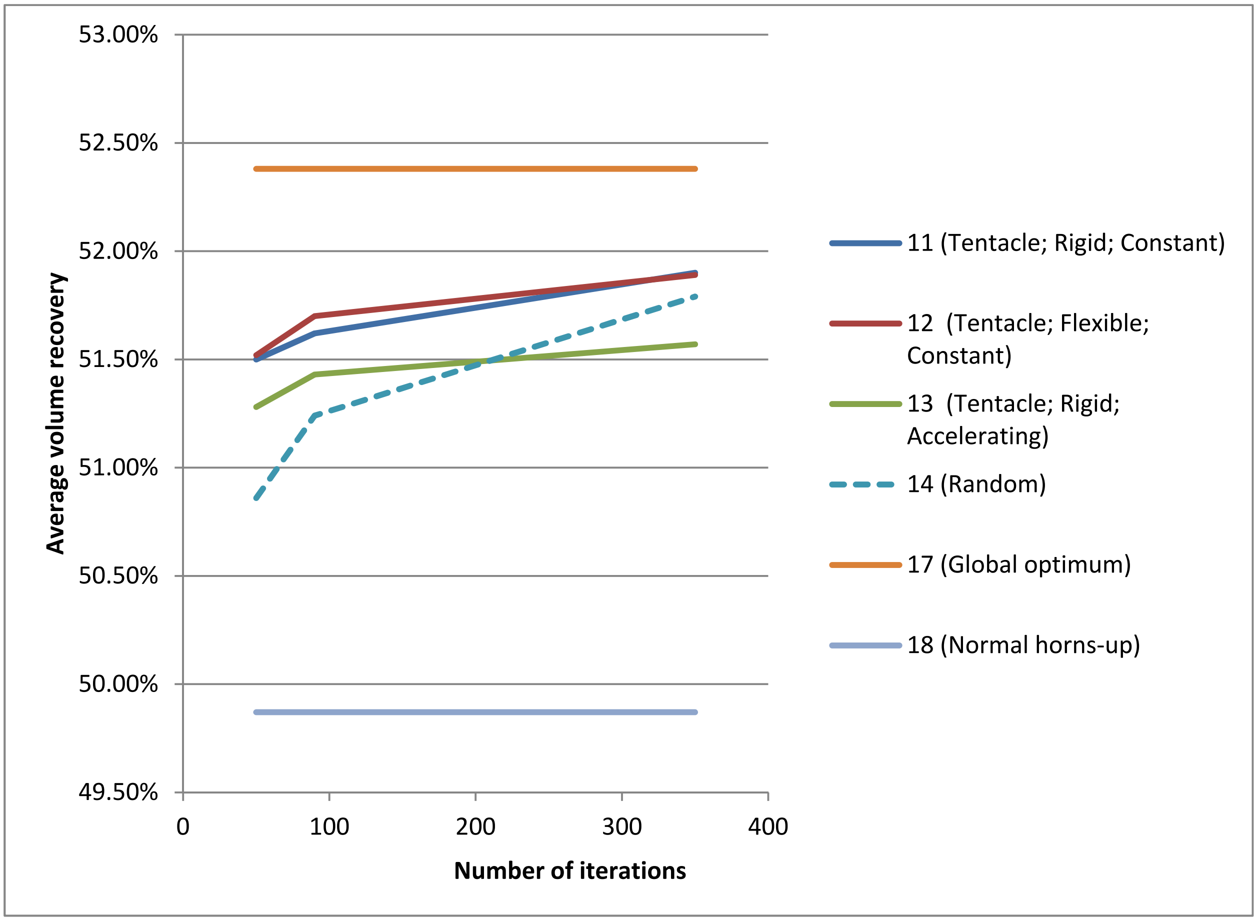
5.2.3 SA



Graph 3: Graphic presentation of SA results

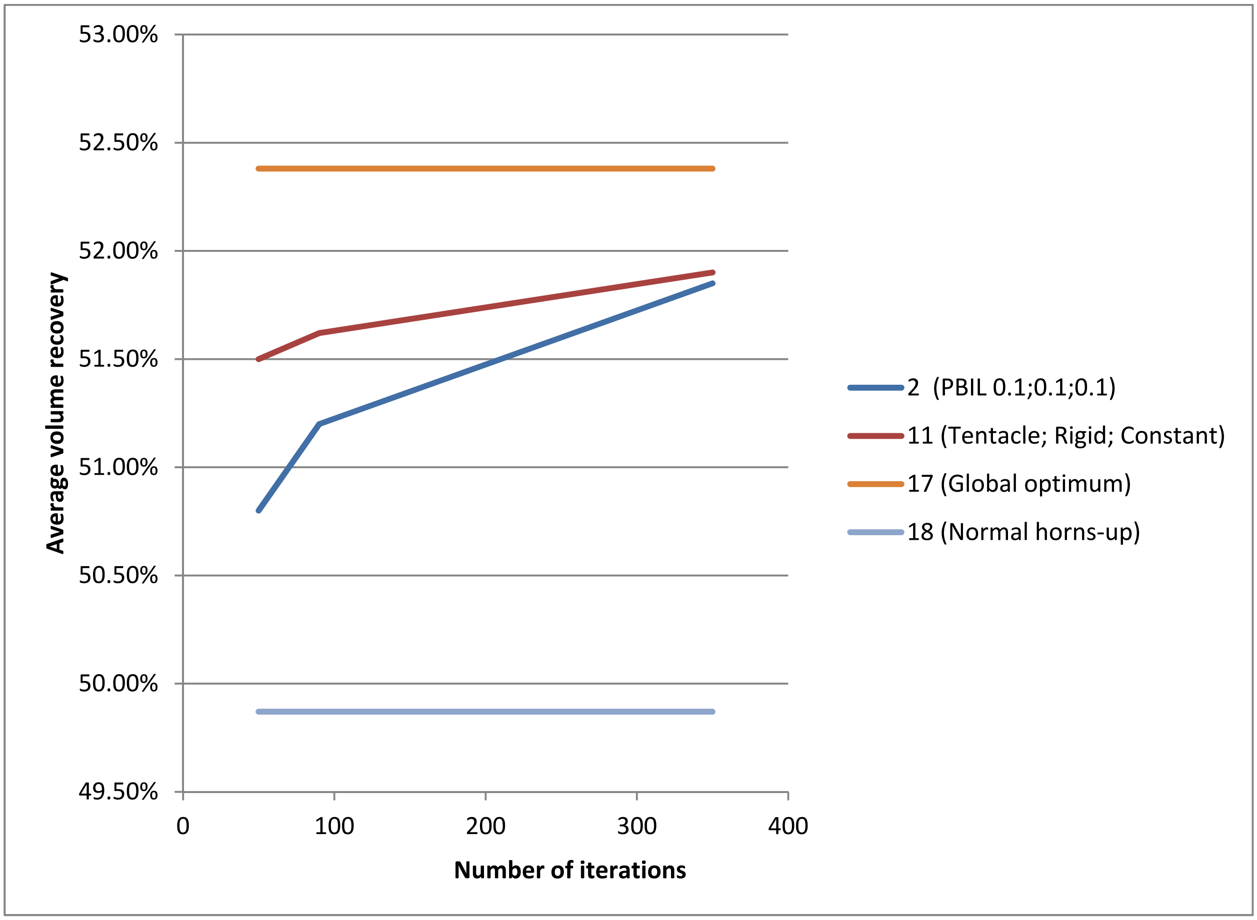
The performance of the SA meta-heuristic is virtually on par with the random search strategy. This algorithm utilized the knowledge of where the good quality area in the search space is, as it started its search in the normal horns-up position. It is possible that the neighbourhood of the focal point was defined too large in this instance. This might allow the focus to move away from the good quality area too quickly, thus minimizing the benefit of knowing where the good quality area is. It is important to note that moves to weaker combinations are very likely at the start of the meta-heuristic, which further negates the benefit of knowing where the good quality area is.

5.2.4 Tentacle algorithm



Graph 4: Graphic presentation of the Tentacle algorithm results

After 90 and 350 iterations, the tentacle algorithm, with constant growth of tentacles, out-performed all other meta-heuristics under consideration. Considering the relative simplicity, it is quite surprising. The knowledge of where the good quality area is, was probably most effectively used in this meta-heuristic. It started its search in the normal horns-up position, which was not the case for the PBIL or PSO meta-heuristics. It also searched around its focal point with a finer resolution than further away from the focal point. It also did not accept any weaker combination in an attempt to leave a potential local optimum. Comparing the performance of the Tentacle algorithm with that of the PBIL, however, (as shown in graph 5) one can appreciate that the PBIL has a higher rate of learning after 350 iterations. This might indicate superior performance from the PBIL with a higher number of iterations.



Graph 5: Average volume recovery per number of iterations for two instances of the above mentioned meta-heuristics

5.3 Further discussions

The knowledge of where good quality areas are in the search space was beneficial in the case of the Tentacle algorithm. In instances where experience of the general location of near optimal results are not present and where the 'optimal' solution can be anywhere in the search space, the benefit of starting in a specific position is lost. In those instances the individual meta-heuristics could be evaluated more accurately. With the exhaustive searching of reduced ranges around the horns-up position, knowledge of where good quality areas are, was also needed.

In Table 9 the difference in average volume recovery between the best and worst performing meta-heuristic after 350 iterations was only 0.84%. With such a small range it should again be emphasized that positioning errors of logs, considered to be small, can cause losses in recovery of 1.06%. (See table 3)

It should also be noted from all the above graphs that the improvement in average volume recovery is of a diminishing nature for all the optimisation strategies considered.

The exhaustive search of reduced ranges yielded 51.74% average volume recovery after 90 iterations and 52.07% after 350 iterations. This outperformed all meta-heuristics under consideration. As mentioned, the knowledge of where the good quality area is, was essential for this strategy. In this context this strategy is considered to be the most suitable.

6. Conclusions

In this empirical study four meta-heuristics were compared in attaining near optimal positioning combinations of logs before breakdown. The PSO meta-heuristic performed best after 50 iterations in this study. After 90 and 350 iterations the tentacle algorithm performed best. A better strategy in all cases was the exhaustive searching of smaller ranges around good quality combinations. This knowledge of where good quality areas are in a search space was also a contributing factor in the tentacle algorithm's effectiveness. In this algorithm the search started at the traditional horns up position and the resolution diminished as the focus shifted away from this position. The good quality area in the search space was therefore searched more thoroughly.

The danger of a poorly selected meta-heuristic was pointed out. The selection of random combinations outperformed some of the meta-heuristics. The potential of using meta-heuristics was however shown. With the best performing meta-heuristic, after 350 iterations, 81% of the potential improvement in volume recovery was attained after evaluating only 9% of the possible combinations. Selecting 9% of the possible combinations randomly will yield only 76% of the potential improvement.

With only 90 iterations; which relates to only 2% of the total number of combinations, the best meta-heuristic attained 73% of the potential improvement compared to 55% attained by random selections.

References

1. Barbour, R.J., Parry, D.L., Punches, J., Forsman, J., and R. Ross. 2003. AUTOSAW Simulations of lumber recovery for small diameter Douglas Fir and Ponderosa Pine from Southwestern Oregon. USDA Research note PNW-RN-543.
2. Bekker, J. 2006. Some meta – heuristics for combinatorial optimisation. Class notes. Department of Industrial Engineering, University of Stellenbosch.
3. Blackman, T. 1999. Older sawmills reap benefits of optimization programs. Wood Technology. August 1999: 34-35.
4. Bowyer, J.L. Shmulsky, R, Haygreen, J.G. 2003. Forest Products and Wood Science: An Introduction. Blackweel Publishing, Iowa.
5. Bryan, E. L. The Best Possible Sawmill: Guidebook for the High-Tech Journey Ahead. San Francisco: Miller Freeman Books, 1996
6. Carino, H.F., Arano, K.G., Blanche, C.A., Patawaran, W.A. 2006. Impact of curve sawing on southern pine dimension lumber manufacturing. Part 1. Lumber volume and value yields. Forest Prod. J. Vol. 56, No. 7/8.
7. Chang, S.J., Cooper, C., and S. Guddanti. 2005. Effects of the log's rotational orientation and the depth of the opening cut on the value of lumber produced in sawing hardwood logs. Forest Prod. J. Vol. 55, No. 10.
8. Chiorescu, S., Grönlund, A. 2000. Validation of a CT-based simulator against a sawmill yield. Forest Prod. J. Vol. 50, No. 6.
9. Cown, D.J., D.L. McConchie, Kimberley, M.O. 1988. Sawing methods for Pinus radiata pruned logs – an indicative study. New Zealand J. of Forestry Science 18(3): 345-58.

10. Del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C., Harley, R.G. 2008. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 12, NO. 2, APRIL 2008 171
11. Denig, J. 1993. Small Sawmill Handbook. Miller Freeman, San Francisco.
12. Faaland, B., and D. Briggs. 1984. Log bucking and lumber manufacturing using dynamic programming. Management Science Vol. 30, No. 2, Feb.
13. FAO [Online]. 2010. FAO forest database. Available at: http://faostat.fao.org/. Accessed: 20 January 2010.
14. Geerts, J.M.P. 1984. Mathematical solution for optimising the sawing pattern of a log given its dimensions and its defect core. New Zealand J. of Forestry Science 14(1): 124-34.
15. Godsmark, R. 2010. The South African Forestry and Forest Products Industry 2009. Forestry South Africa presentation. Available at: www.forestry.co.za. Accessed 12 November 2010.
16. Hallock, H., A.R. Stern, and D.W. Lewis. 1976. Is there a “best” sawing method? USDA forest service research paper FPL 280.
17. Hamner, P., White, M., Araman, P. 2005. Cant, part volume little affected by sawing method, study shows. Pallet Enterprise, May 2005.
18. Ingber L. 1989. Very fast simulated re-annealing. Mathematical and Computer Modelling 12(8): 967-973.
19. Kennedy, J., Eberhart, R.C. 2001. Swarm Intelligence. Morgan Kaufmann, San Francisco.
20. Lewis, D.W. 1985. Best opening face system for sweepy, eccentric logs. A user’s guide. USDA, For. Science, For. Prod. Lab.
21. Malan, F.S. 1988. Relationships between Growth Stress and some tree characteristics in South African grown Eucalyptus grandis. South African Forestry J. 144: 43-46.

22. Maness, T.C., and D.M. Adams. 1991. The combined optimisation of log bucking and sawing strategies. Wood and Fiber Science, 23(2), Pages: 296-314.
23. Maness, T.C., and W.S. Donald. 1994. The effect of log rotation on value recovery in chip and saw sawmills. Wood and Fibre Sc. 26(4), Pages: 546-555.
24. Mendoza, G.A., and B.B. Bare. 1986. A two-stage decision model for log bucking and allocation. Forest Prod. J. Vol. 36, No. 10.
25. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. 1953. Equation of state calculations by fast computing machines. The Journal of Chemical physics, Vol. 21, No. 6, June 1953.
26. McAdoo, J.C. 1969. Computer simulation of small log processing. Forest Prod. J. Vol. 19, No. 4.
27. Michalewicz, Z., Fogel, D.B. 1998, How to solve it: Modern Heuristics. Springer-Verlag Berlin.
28. Monserud, R.A., Parry, D.L., Todoroki, C.L. 2004. Recovery from sawn logs with sweep. New Zealand J. of Forestry Science 34(2).
29. Occena, L.G. and Tanchoco, J.M.A. 1988. Computer graphics simulation of hardwood log sawing. Forest Prod. J. Vol. 38, No. 10.
30. Pedersen, M.E.H., Chipper, A.J. 2009. Simplifying Particle Swarm Optimization. Applied Soft Computing. Volume 10, Issue 2, March 2010, Pages 618-628.
31. Peter, R.K., Bamping, J.H. 1962. Theoretical sawing of pine logs. Forest Prod. J. Vol. 12, No. 11.
32. Pinto, I., Knapic, S., Perreira, H., Usenius, A. 2006. Simulated and realised industrial yields in sawing of maritime pine (Pinus pinaster Ait.). Holz als Roh- und Werkstoff 64: 30-36.
33. Reinders, M.P., Hendriks, Th.H.B. 1989. Lumber production optimization. European Journal of Operational Research 42, 243-253.
34. Rayward-Smith, V.J., Osman, I.H., Reeves, C.R., Smith, G.D. 1996. Modern Heuristic Search Methods. Wiley, New York.

35. Schultz, H. 1993. The development of wood utilisation in the 19th, 20th, and 21st centuries. Forestry Chronicle 69(4): 413-418.

36. Shi, Y., Eberhart, R.C. 1998. A modified particle swarm optimizer. In Proceedings of 1998 IEEE International Conference on Evolutionary Computation. International Conference on Evolutionary Computation. ICEC'98. Anchorage, AK. USA. 04-09 May 1998. pp. 69-73.

37. Smit, N.H. 2009. Particle swarm optimisation for log positioning. Unpublished honours project report, Department of Forest and Wood Science, Stellenbosch University.

38. Steele, P.H., Wengert, E.M., Little, K. 1987. Simplified procedure for computing best opening face position. Forest Prod. J. Vol. 37, No. 5.

39. Thomas L., Lamine Mili L., Shaffer A.C., Thomas E. 2004. Defect detection on hardwood logs using high resolution threedimensional laser scan data. International Conference on Image Processing : (ICIP 2004) (Singapore, 24-27 October 2004)

40. Todoroki, C.L., Rönnqvist, E.M. 1999. Combined primary and secondary log breakdown. Journal of the Operational Research Society. Vol. 50.

41. Todoroki, C.L., Rönnqvist, E.M. 2002. Dynamic control of timber production at a sawmill with log sawing optimisation. Scand. J. For. Res. 17: 79-89.

42. Vuorilehto, J. and T. Tulokas. 2007. On log rotation precision. Forest Prod. J. Vol. 57, No. 1/2.

43. Walker, J.C.F. 1993. Primary Wood Processing – Principles and Practice. Chapman Hall, London.

44. Wang, S.J., and Giles, D.R. 1989. Effects of various factors on computer-optimized bucking system performance. Forest Prod. J. Vol 39, No. 11/12.

45. Wang, S.J., B.D. Munro, D.R. Giles, Wright, D.M. 1992. Curve sawing performance evaluation. Forest. Prod. J. Vol. 42, No. 1.

46. Wessels, C.B., Price, C.S., Turner, P., Dell, M.P. 2007. Integrated harvesting and sawmill operations using an optimised sawmill production planning system. In Ackerman, P.A., Langin,

D.W., and M.C. Antonides (eds), 2006, Proceedings of the International Precision Forestry Symposium, Stellenbosch University, South Africa, 5-10 March 2006. ISBN 0-7972-1121-7.

47. Wessels, C.B. 2008. Unpublished data on PBIL optimisation for log positioning. Department of Forest and Wood Science, Stellenbosch University.
48. Wessels C.B. 2009a. Wood Science class notes. University of Stellenbosch.
49. Wessels C. B. 2009b. Cant sawing log positioning optimisation: A simulation study. Forest Prod. J. Vol. 59, No. 4.
50. Wessels, C.B., Price, C.S., Turner, P., Dell, M.P. 2006. Integrating harvesting and sawmill operations using an optimized sawmill production planning system. In Ackerman, P.A., Langin, D.W., and M.C. Antonides (eds), 2006, Proceedings of the International Precision Forestry Symposium, Stellenbosch University, South Africa, 5-10 March 2006. ISBN 0-7972-1121-7.
51. Williston, Ed M. 1988. Lumber manufacturing - The Design and Operation of Sawmills and Planer Mills. Miller Freeman, San Francisco.

Appendix A: Proposal for a novel log positioning system

Technology Advancement Programme 2006: Pre-Proposal

Focus Area: Advanced Manufacturing / ICT

Log-Positioning Optimisation System

Project Co-ordinator:

Brand Wessels

Stellenbosch University

Tel (w): 021 808 3319

Cell: 083 778 0517

Fax: 021 808 3603

E-mail: cbw@sun.ac.za

Postal address: Private Bag X1, Matieland, 7602

Physical address: Paul Sauer Building, Bosman St., Stellenbosch, 7600

Alternative contact person:

John Mortimer

Stellenbosch University

Tel (w): 021 808 3806

Cell: 082 891 9960

Fax: 021 808 3603

E-mail: jhm@sun.ac.za

Postal address: Private Bag X1, Matieland, 7602

Physical address: Paul Sauer Building, Bosman St., Stellenbosch, 7600

Consortium members:

Stellenbosch University

Nukor

Multisaw

Submission date: 27 March 2006

EXECUTIVE SUMMARY

The past two decades have been marked by severe difficulties for the South African solid wood processing industry. A total of 143 out of 188 formal sawmills in South Africa have closed down since 1988. This industry, however, is still an important employment source, providing direct jobs to about 113 000 people (the number of employees can be seen in proper perspective when it is compared with the approximately 112 000 people directly employed by the more prominent automotive manufacturing industry). If the trend in sawmill closures is to be reversed in the long run, technological solutions will have to play an important part in improving the efficiency of our industry.

One of the most critical areas in the timber production process in terms of the conversion efficiency is the positioning of a round log in front of the primary breakdown saw. This proposal describes the development of a system that optimises the positioning of a log in front of the primary breakdown saw. The system uses an innovative optimisation model and log-marking system, together with three-dimensional laser scanning, to replace the more subjective operator judgement when positioning a log. Unlike other log optimisation systems currently in use, no expensive automated positioning equipment is required with this system. The product addresses the requirement of improved log conversion efficiency by optimising the log-positioning process without the use of expensive automated positioning equipment. The total budget for the project is R5,5 million over three years.

The product is aimed at the low- to medium-sized sawmills that cannot economically justify existing log-positioning optimisation systems – the market segment in which nearly all South Africa's formal sawmills fall. The unique optimisation concept will enable sawmills to use their existing operator-controlled log-positioning equipment to achieve optimum positioning results. Apart from the fact that a much smaller capital outlay is required to install the system, simulation results indicate that the mathematical optimisation model that was developed will be able to provide superior solutions to those provided by currently available systems. The same conditions found in the South African timber-processing industry are also present in many other parts of the world, facilitating the international marketing of the product.

This project aims to develop an innovative technology that will significantly impact on our industry and similar industries overseas, and which will help to further establish our small sawmilling equipment manufacturing sector.

(The values quoted in this summary are from: Crickmay Associates. 2004. *Supply and Demand Study of Softwood Sawlog and Sawn Timber in SA*; and International Organisation of Motor Vehicle Manufacturers. 2004. *The World's Automotive Industry: Some Key Figures.* OICA 4 Volets (http://www.oica.net).

1. DESCRIPTION OF RESEARCH AND DEVELOPMENT IDEA / INNOVATION

The sawmilling industry in South Africa has endured severe difficulties over the last two decades, experiencing closure of nearly 80% of all formal sawmills in the country. The closures have had devastating social consequences for the rural areas, where forestry and sawmilling are often the only formal means of employment available. One of the main problems experienced has been the conversion inefficiencies in medium-sized sawmills, which still tend to make use of old, outdated equipment. Due to the geographical spread and ownership patterns of forest resources in South Africa, in most cases large high-throughput sawmills, which might otherwise justify implementation of the latest, most efficient processing technologies, are not viable. The log-positioning optimisation system proposed here is an attempt to develop a system that will be capable of drastically improving the conversion efficiencies of small to medium-sized sawmills.

A measure of the processing efficiency in a sawmill is the value recovery that it achieves – that is, the average final product Rand value created per cubic metre of round log processed. Closely related to value recovery is a measurable called volume recovery, which can be defined as the product volume output as a percentage of log volume input. Maximising these two measurables is the management objective of most sawmills worldwide.

A detailed sawmill simulation study (conducted by C.B. Wessels in 2004) has already quantified the significant improvements in value and volume recovery possible if a log is positioned optimally in front of the primary breakdown saw of a sawmill. The objective of this project is to design and develop an innovative log-positioning system aimed at both the South African and the international sawmilling industries. The proposed system aims to be more efficient in terms of value and volume-recovery solutions than are other log-positioning systems that are currently available. Due to a unique optimisation approach, the proposed system will also be much less costly than current systems. Initial cost analyses and simulation studies suggest that the envisaged goals may be attained if a few novel system components are successfully developed. The proposed system will consist of the following components:

1. **A 3D laser-scanning system; a frame; and the materials handling component.** The scanning system must provide an accurate three-dimensional image of a log. Three-dimensional log-scanning is a well-developed technology, and scanning equipment available from Microtec will be used. The scanning cameras and log-marking system will be mounted on a single structure that will be placed over a conveying system that feeds logs into a sawmill. The development of log-positioning equipment using limited automation, together with radio frequency tags, will also be investigated.
2. **Process optimisation software.** Process optimisation software is a critical component of the system and will determine the degree of efficiency in terms of value and volume recovery of which the system will be capable. Such software makes use of three-dimensional images

obtained from the scanning system, as well as information on the sawmill manufacturing process, to determine the optimum log position. The basic problem with finding an optimum log position is that, though literally thousands of different positions are possible for any one log, the optimum position must be found within only a few seconds. A unique log-positioning optimisation process, using a simulation system coupled with a meta-heuristics algorithm, has already been tested in an initial study (C.B. Wessels, 2004). Very positive results were obtained in the study, showing excellent possibilities for improvement when using the proposed method when compared with other traditional positioning methods.

3. **The log-marking system.** An innovative approach applicable to especially low to medium-volume throughput sawmills is proposed for positioning and feeding logs through the primary breakdown saw. Automated positioning and feeding systems normally used in log-positioning optimisation systems tend to be extremely costly, so that they are only truly applicable in high-throughput sawmills. In the automated systems, scanning takes place immediately in front of the automated positioning equipment. In the proposed system, however, the log will be scanned at a separate location, allowing for marking of the log ends. The existing operator-controlled positioning equipment in a sawmill can then be used to position the log according to the marks already made, obviating any need for expensive positioning equipment. An additional advantage of the system is that log movement during scanning and positioning, which is a major problem in the high throughput automated positioning systems, will be eliminated. An appropriate log-marking system will, however, have to be developed for operation together with the scanning system. Three different marking methods will be evaluated before development, namely a radio frequency identification method, an automated paint-mark application method, and manual marking.

Innovation and unique features

The major innovatory feature of the proposed product is the coupling of the simulation approach to a meta-heuristics algorithm in order to secure an optimal position solution, which is aimed at providing superior results to those obtained by means of the methods currently in employ. The optimisation method, which is based on a genetic algorithm, facilitates much wider searching (in terms of log-positioning variables) in order to secure an optimum position. Testing by means of a simulation study has shown that the proposed method is capable of providing significant improvements on the existing methods.

The second unique feature of the system entails the log-marking system employed. The system allows sawmills to use existing operator-controlled positioning equipment to position a log under the direction of a laser beam according to marks made on the log (thus obviating any need for automated positioning equipment). As the proposed log-positioning optimisation system will be feasible for small to medium-sized sawmills to implement, it holds significant cost-saving implications for the industry. An added advantage of the log-marking system is that log movement during transport will not adversely affect results. Current automated log-positioning systems tend to record log images inaccurately when logs

The use of Internet search engines to locate relevant information

A Boolean search (using the operator "AND") was used on the search engines www.dogpile.com and www.ananzi.co.za to locate websites featuring the words "sawmill", "log", "optimisation / optimization", "positioning", and "scanning" in combination. Although 62 sites in all were located in this way, none of the products or systems described on the websites employed the optimisation approach proposed in this project. An interesting technological review of the Canadian Federal Government Department of Industry (http://strategis.ic.gc.ca/epic/internet/infi-if.nsf/en/fb01381e.html) highlighted problem areas and opportunities for research relating to the forest industry. The following information relating to problems experienced during the primary breakdown process in the average sawmill appears on the site, which was last updated in April 2005:

"Computing and software capabilities have to be improved to ensure that the optimisation process is completed before a solution is retained."

"The weak link in the primary breakdown process is the mechanical components that must implement breakdown solutions identified by the optimisation software. Mills currently have to accept log movement at every step of the log positioning and breakdown process"

The current project aims to solve both problems.

Existing patents located by means of a search conducted on Espacenet

A combination of the keywords "log", "saw" and "positioning" to search for existing patents and patent applications on the www.espacenet.com website resulted in finding 25 patents or patent applications described in these terms in their abstracts or titles. Addition of more keywords resulted in a zero outcome. No patents exist for optimisation algorithms. Most of the patents located related to a combination of automatic positioning equipment and scanning methods, none of which is of relevance to the proposed project.

The sharing of intellectual property and patents

Intellectual property will be shared between the consortium members based on the inputs received from each.

2. CONSORTIUM / ORGANISATIONAL PLAN

The responsibilities and core activities of Consortium members

Forest- and forest-products-related research and development forms one of the two main activities of the Department of Forest and Wood Science at **Stellenbosch University**. As such, this project will definitely form part of the core activities within the Department. The Forest and Wood Science Department will be the leading organisation in this project, providing overall project management. The Department will also be responsible for development of the relevant process optimisation software. The Mechatronics Group, based in the Mechanical Engineering Department at Stellenbosch University, specialises in the design of integrated mechanical and electronic systems. The Group will be responsible for the design and integration of the scanning, material-handling, and log-marking system components.

Multisaw is a South African manufacturing company specialising in the design, development and manufacturing of wood-processing equipment. Multisaw, which was first established in 1991, has been responsible for introducing a number of new products into the South African market since then. Due to its experience with manufacturing sawmilling equipment, Multisaw will manufacture the commercialised product concerned. Since the company is well acquainted with the sawmilling environment, it will also be intimately involved with all the design stages of the proposed product. In terms of the National Small Business Amendment Bill of 2003, Multisaw is regarded as a small business (the confidential financial statements of Multisaw are available for scrutiny by Innovation Fund personnel).

Nukor is the largest seller of sawmilling equipment in Africa. Apart from the core business of marketing and providing maintenance support for mostly European sawmilling equipment, Nukor also has an engineering division, so that it can design and manufacture some of its own equipment. As such, Nukor will be responsible for the commercialisation and marketing of the relevant positioning optimisation equipment. The company will also be involved with the design of the proposed product. In terms of the National Small Business Amendment Bill of 2003, Nukor is regarded as a medium-sized business (the confidential financial statements of Nukor are available for scrutiny by IF personnel).

Risk sharing

Most of the research and development work preceding this proposal has been conducted at the Department of Forest and Wood Science at Stellenbosch University over the past two years. The investment to date includes various simulation studies, the development of an optimisation algorithm, and industry visits, the value of which could roughly convert to R250 000. For the development and testing of components of the proposed system, use will largely be made of already existing equipment and facilities available at Stellenbosch University. As part of sharing the risk of the project, Nukor will not charge a fee for the time spent by their employees on this project.

Organisational plan

The project can be separated into four components, with a specific organisation being responsible for each component. The design and development of a laser-scanning system and a log-marking system, as well as materials handling, will be performed by the Mechatronics Group at Stellenbosch University, while the design and development of the process optimisation software will be undertaken by the Department of Forest and Wood Science of Stellenbosch University. Nukor will be responsible for the commercialisation of the product. While Multisaw and Nukor will both be involved in all the design stages, Multisaw alone will be responsible for the manufacture of the commercialised product. Overall project management will be performed by the Department of Forest and Wood Science at Stellenbosch University.

A project management committee consisting of at least one representative of each organisation concerned will manage the project. Decision making will take place by way of consensus. However, when necessary, voting will occur, with each organisation having one vote. Stellenbosch University's financial system will be used for purposes of project finance management.

Key management personnel

Brand Wessels holds a BEng (Industrial Engineering), as well as an MSc *cum laude* (Wood Science) degree. While working for five years for Environmentek (CSIR) as a project manager focusing on solid-wood processing research and development, he was responsible for, amongst other projects, the design, development and commercialisation of the sawmill simulation software package Simsaw 6, as well as of the linear-programming-based Sawmill Production Planning System (SPPS) software package. His focus in his current position with the Department of Forest and Wood Science at Stellenbosch University is on solid-wood processing. Wessels will be responsible for the design and development of the process optimisation software, as well as for the overall management of the project.

John Mortimer, a forestry graduate, worked for many years in the forestry and sawmilling industries in South Africa and Canada, including as the national production manager and marketing manager of Mondi Timber. After serving as the national chairman and, later, the executive director of the South African Lumber Millers Association (SALMA) for 9 years, he joined Stellenbosch University as a faculty manager in the Faculty of Agricultural and Forestry Sciences. John is the current national chairman of the South African Wood Preservers Association (SAWPA), as well as the chairman of the Board of Trustees of the Furniture Technology Centre (Furntech). He stands to play an important role in the commercial management of the project. His knowledge and experience of the sawmilling industry will be especially useful during the initial design phase, as well as during the commercialisation phase.

Kristiaan Schreve holds a PhD in Mechanical Engineering from Stellenbosch University. After working as a production manager in the GCC Rapid Product Development laboratory, he assumed a senior lectureship with the Mechanical Engineering Department of Stellenbosch University in 2003. His involvement with the Mechatronics and Design Group in the Mechanical Engineering Department revolves around his concern with the fields of reverse engineering, CAD modelling, and rapid prototyping. Schreve's experience with product design and development is likely to prove crucial throughout the project.

Corné Coetzee, with a PhD in Mechanical Engineering, is currently a senior lecturer in the Mechanical Engineering Department at Stellenbosch University. Apart from his experience of the design and optimisation of specifically mining equipment and his focus on computational mechanics research, his fields of expertise include those of strength of materials and finite element methods. With his wide-ranging experience, Coetzee will be responsible for certain of the mechanical design tasks that form an indispensable part of the project.

Stephen Röth, a director at Nukor, holds a BSc Wood Science degree, as well as an MBA degree. His 17 years experience in the sawmilling equipment supply and manufacturing industry includes his completion of numerous multimillion Rand wood-processing equipment installations in various parts of the world. As well as being responsible for the commercialisation of the product, he will also be involved in the design stages of the project.

Neil Murray is the owner and manager of Multisaw, which he started by himself in 1991. Multisaw has been responsible for developing and commercialising various new sawmill processing products. Apart from managing the company, Murray is also involved in the design of new products. As well as being involved with the design of the log-positioning system, he will also be responsible for manufacturing the commercialised product.

3. MARKETING PLAN / COMMERCIALISATION

Market opportunity

Mounting financial pressures are currently forcing wood processors to increase their efficiency in converting logs into lumber. For a medium-sized South African softwood sawmill processing 100 000 m^3 of round logs per year, an increase in volume recovery of 1% translates into roughly R1,5 million extra income, without the need to use more raw materials. The importance of volume and value recovery efficiency in wood-processing industries is accepted worldwide and is usually one of the primary management measures of any sawmill. From an environmental perspective, increasing log-conversion efficiency is of equal importance, as the efficient use of scarce timber resources should make a significant contribution to helping the whole world move towards a more sustainable supply-demand scenario.

The main objective of the proposed log-positioning optimisation system is to increase the value and volume recovery efficiency of the log breakdown process. A simulation study has already shown that the proposed optimisation method will result in an increase in volume recovery of at least 2% to 5% when replacing operator judgment in a typical South African sawmill. Due to the similarity with other softwood log breakdown patterns in most other countries, the results stand to have universal implications.

Currently available automated log-positioning optimisation equipment is aimed at satisfying the needs of high-throughput sawmills. However, a sizable proportion of sawmills fall outside the required throughput size, failing to justify the purchasing of the extremely expensive equipment. In South Africa, only two sawmills have so far implemented automated log-positioning optimisation equipment, causing doubt as to whether other sawmills will be able to justify purchasing such a system at the current price. As previously mentioned, current systems also experience difficulty with log stability, resulting in scanning inaccuracy and sub-optimum positioning.

Considering the minimum volume recovery increase scenario of 2% for a South African sawmill of

100 000 m³ annual log intake, the overall increase should translate into an extra R3 million income per annum. Such an increase in volume recovery will lead to negligible extra cost, so that the extra income will be able to be directly added to the company's bottom line. Due to the unique positioning optimisation approach, no expensive automated positioning equipment will be required. Initial cost calculations suggest that manufacture of the product should cost less than R800 000. At an estimated purchase price of R1,2 million, a 100 000 m³ throughput sawmill stands to benefit from a payback period of under six months.

The key success factors of the product are its relatively low cost (a short payback period); enhanced efficiency (an increase in the relevant sawmill's volume and value recovery); and superior quality. The need for equipment to be robust and easy to maintain in the sawmilling environment will form a primary focus of the design process. Another very important advantage of the product is that it does not interfere with the production process. If it breaks down or while installation takes place, production can continue normally albeit without the optimisation option.

Market segment analysis

The following scenario will be used to estimate the potential market size for the proposed product: A one-year payback period is required; the product will result in a volume and value recovery increase of 2%; the average selling price of timber is R1 500/m³; and the selling price of the product is R1,2 million. Based on these very conservative values, the product will be affordable for a sawmill with 40 000 m³ log throughput.

Currently, 41 sawmills of 40 000 m³ or higher annual log throughput exist in South Africa. Internationally, the exact number of sawmills is not known. However, based on the fact that South Africa currently produces 0,4% of the world's saw-timber logs, the potential international market size is likely to be more than 1 000 sawmills. However, the incompatibility of the proposed product with certain types of log breakdown equipment, such as with carriage bandsaw systems, is likely to reduce the actual potential market size by an unknown percentage. This relatively minor shortcoming is unlikely to have a significant impact on the local market, however, as nearly all South African sawmills do, in fact, use production systems compatible with the proposed positioning optimisation system.

(The values given in the "Market segment analysis" above are based on information obtained from the following sources: SA Lumber Index, September 2005, Crickmay and Associates; Commercial Timber Resources and Roundwood Processing, 1996/7, the Department of Water Affairs and Forestry; Personal Communication, Mandy Gibson, Crickmay and Associates, November 2005.)

Competitor analysis

The proposed product is aimed in the first place at a segment of the market that cannot justify the expensive automated positioning optimisation equipment currently available from European and North American manufacturers. As no other product presently obviates the need for expensive automated positioning equipment, in the low to medium-throughput sawmill market segment, no direct competition exists from other products at the moment.

The competitive advantage of the proposed product over current systems is based on the following factors:

– the significantly lower cost, due to the unique optimisation approach;

– the optimisation algorithm providing optimum or close-to-optimum positioning solutions;

– the non-interference of the scanning and marking operations with the production line, resulting in the absence of production downtime, even in cases where the scanning and marking system temporarily fails (which is of particular importance in the sawmilling environment, which is often situated at some distance from a service centre); and

– the simplicity and robustness of the system, which makes it relatively easy to maintain and repair.

Marketing strategy

The marketing strategy will involve initially focusing on a few successful installations in South Africa. Since the country has a small and close-knit sawmilling community, initial successful installation of the proposed system at a local sawmill will be critical. The consortium member Nukor, which will assume responsibility for marketing the product, has an excellent reputation not only in South Africa, but also throughout other parts of the world, and has established close contact with many sawmills on the continent. The Capesaw sawmill in Stellenbosch has already assented to making its plant available for testing of the prototype.

A similar commercial and technological environment as that present in South Africa is also present in many other parts of the world, which will facilitate international marketing of the product. The Ligna Fair that takes place regularly in Germany is seen as providing the most suitable venue for international exposure of the proposed product, as the Fair is the largest wood-processing equipment fair in the world. Of the approximately 100 000 worldwide visitors to the Fair, 65% tend to be the purchasing decision makers in their respective companies. However, international marketing of the product will only follow the first few successful installations of the systems in South Africa and will, therefore, not fall within the time-scope allowed for this project.

Stephen Röth of Nukor, assisted by John Mortimer of Stellenbosch University, will be responsible for the commercialisation stage of the project. Nukor, as the largest sawmilling equipment retailer in South Africa, already serves as the agent for many of the foremost sawmilling equipment brands throughout the world. Röth and Mortimer both have many years of management experience in the sawmilling and sawmill equipment marketing environments respectively.

4. PROJECT ACTION PLAN AND 5. BUDGET

The attached Gantt chart, including the budget, explains the different tasks involved in the project, the cost per task, the timeframe and set deadlines, the breakdown of costs per consortium member, as well as the anticipated costs per quarter. .All costs are, where applicable, VAT inclusive. Buffer-times, which have been included in the estimation of tasks, should ensure that no slippage occurs. Regular project meetings will ensure the timely identification of tasks at risk of late completion and allow for the taking of corrective actions already planned. Each major task will culminate in the reaching of a significant milestone (see tasks 1 – 6 of the attached Gantt chart). Any of the milestones suggested could act as a potential exit point for the project if an insurmountable problem arises during the project.

6. PROJECT RISKS

The current opportunity that exists in the market for a log-positioning optimisation system aimed at low to medium-sized sawmills being filled by a competitive product is a major risk for the project envisaged. A countermeasure to this risk involves proceeding as fast as possible with product development. Excellence in, and innovation of, design will also increase the value of the product in relation to that of other possible future competitive products.

In order to overcome the risk attendant on having to adhere to prescribed time and budget constraints, an experienced project manager will be involved from the start in the planning and execution of the project. Stellenbosch University's financial and project management system will also be used to keep track of project expenses. Apart from regular technical project meetings and ongoing communication between members of the consortium, project management meetings will be held every six months. The objective of these meetings will be to evaluate the project in terms of budget, time and scope and to plan corrective actions, when, and if, necessary.

Technical risks relate to the accuracy and robustness of the system in the sawmilling environment. To countermeasure such risks, members of the engineering design team will visit several sawmills in South Africa prior to commencement of the design phase. Measurements of vibration, light conditions and first-hand experience of the sawmilling environment should then help the engineers involved to quantify the design parameters of the project. The involvement of an experienced sawmilling manager and sawmill equipment manufacturers in the design stages will further ensure that the product complies with industrial requirements. Testing of components in the sawmilling environment will also take place as prototypes of the different components near completion.

Appendix B: Results Summary

Summary of results after 350 iterations

Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Simulated annealing results - Linear cooling: Max recovery after 350 iterations	Simulated annealing results - High starting temperature: Max recovery after 350 iterations	Simulated annealing results - Standard parameters: Max recovery after 350 iterations	Tentacle Algorithm - Accelorated growth: Max recovery after 350 iterations	Tentacle Algorithm - Flexible: Max recovery after 350 iterations	Tentacle Algorithm - Standard: Max recovery after 350 iterations
Avg.		49.872	52.378		51.797	51.810	51.818	51.568	51.893	51.897
1		35.86	43.32		41.68	40.99	41.21	40.29	42.38	40.29
2		41.51	43.78		42.91	42.85	43.32	42.85	43.32	42.85
3		50.36	53.66		51.85	52.44	53.47	52.3	51.91	51.91
4		45.7	46.29		46.09	45.7	46.29	46.09	46.09	46.09
5		43.73	44.06		43.73	43.73	44.06	44.06	44.06	44.06
6		48.84	51.41		51.12	51.12	50.84	51.12	51.12	51.12
7		43.93	46.68		46.13	46.68	45.13	46.68	45.88	46.68
8		45.68	46.81		45.93	46.68	46.81	46.81	46.81	46.81
9		45.44	48.09		47.87	47.2	47.32	46.76	46.76	46.76
10		68.54	71.09		70.42	71.09	71.09	70.86	70.86	70.86
11		51.61	53.55		53.16	53.16	52.97	53.35	53.35	53.35
12		48.7	52.97		51.61	51.61	51.22	51.03	51.61	51.03
13		44.61	47.85		46.79	46.79	45.26	46.55	44.61	47.2
14		44.86	45.96		45.96	45.96	45.41	45.41	45.41	45.96
15		45.68	49		48.31	48.31	48.72	48.17	48.31	48.72
16		47.05	48.48		47.41	47.41	48	48.12	48.12	48.12
17		39.02	42.69		42.48	42.48	42.69	42.48	42.48	42.48
18		47.84	50.25		50.25	50.25	49.1	50.25	49.62	49.94
19		40.03	43.66		43.29	43.29	43.48	42.36	42.83	43.66
20		45.06	47.3		46.46	46.46	45.99	47.11	47.2	47.2
21		50.22	51.62		50.22	50.22	50.78	50.22	50.64	50.64
22		51.76	54.99		54.43	54.43	54.71	52.45	54.99	54.71
23		54.88	58.17		57.47	57.47	57.82	55.82	58.17	57.94
24		54.59	56.7		56.2	56.2	56.5	55.09	56.7	56.7
25		57.11	58.92		58.52	58.52	58.52	57.11	58.81	58.92
26		48.03	51.62		50.48	50.48	51.62	48.21	51.62	50.74
27		50.64	53.26		52.33	52.33	51.87	50.94	52.02	52.02
28		54.72	57.5		57.04	57.04	56.81	55.11	57.35	57.35
29		46.89	47.58		47.16	47.16	47.58	46.89	47.58	47.58
30		51.88	54.23		53.76	53.76	54.23	52.37	54.17	53.76
31		50.29	52.24		51.75	51.75	51.87	51.51	51.63	51.75
32		49.81	52.24		51.99	51.99	51.51	51.14	51.26	51.14
33		51.27	53.1		52.49	52.49	53.1	53.1	52.08	53.1
34		46.45	48.81		48.03	48.03	48.81	48.81	48.81	48.81
35		49.08	52.75		51.7	51.7	51.88	51.35	51.88	51.53
36		47.98	49.35		49.35	49.35	48.43	49.19	49.19	49.19
37		52.71	53.11		52.71	52.71	52.98	52.71	53.11	53.11
38		54.05	56.61		55.6	55.6	56.34	56.21	56.34	56.54
39		50.61	52.64		52.64	52.64	52.28	52.4	52.4	52.4
40		55.28	56.6		55.88	55.88	56.12	56.6	56.6	56.6
41		46.27	53.4		52.84	52.84	53.4	52.32	52.32	52.32
42		49.98	55.03		54.47	54.47	52	51.99	52	52
43		54.56	54.87		54.56	54.56	54.56	54.87	54.87	54.87
44		49.47	52.77		52.5	52.5	52.63	52.5	52.5	52.5
45		39.81	46.87		46.87	46.87	45.69	46.79	46.79	46.72
46		56.28	57.22		57.04	57.04	57.16	56.99	56.93	57.16
47		57.04	57.98		57.98	57.98	57.66	57.72	57.72	57.72
48		58.93	60.36		59.29	59.29	60.36	60.36	60.36	60.36
49		53.8	54.83		54.69	54.69	54.6	54.6	54.6	54.6
50		54.79	54.79		54.79	54.79	54.79	54.79	54.79	54.79
51		55.83	59.94		57.68	57.68	59.44	57.66	59.94	57.66
52		52.58	53.99		53.85	53.85	53.85	53.85	53.85	53.85
53		51.53	52.58		52.58	52.58	52.16	52.58	52.58	52.58
54		50.97	53.27		52.9	52.9	53.27	53.27	52.54	52.54
55		51.94	52.72		52.72	52.72	52.3	52.3	52.3	52.72
56		52.73	55.48		55.48	55.48	54.43	54.96	54.96	54.96
57		57.48	58.61		58.61	58.61	58.43	58.43	58.43	58.43
58		52.13	55.52		53.49	53.49	54.15	55.52	55.52	55.52
59		52	53.69		52.71	52.71	52.71	52.96	52.77	53.19
60		41.87	49.79		49.58	49.58	49.33	49.71	49.71	49.71

Summary of results after 90 iterations

					Simulated annealing results - Linear cooling	Simulated annealing results - High starting temperature	Simulated annealing results - Standard parameters	Tentacle Algorithm - Accelorated growth	Tentacle Algorithm - Flexible	Tentacle Algorithm - Standard
Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 90 iterations	Max recovery after 90 iterations	Max recovery after 90 iterations	Max recovery after 90 iterations	Max recovery after 90 iterations	Max recovery after 90 iterations
Avg.		49.872	52.378		51.146	51.132	51.238	51.425	51.705	51.622
1		35.86	43.32		39.12	37.26	40.05	40.29	38.66	40.29
2		41.51	43.78		42.91	42.85	42.85	42.85	43.32	42.85
3		50.36	53.66		51.67	52.44	51.81	51.91	51.91	51.91
4		45.7	46.29		46.09	45.7	45.7	46.09	46.09	46.09
5		43.73	44.06		43.73	43.73	43.73	44.06	44.06	44.06
6		48.84	51.41		49.84	51.12	50.84	51.12	51.12	51.12
7		43.93	46.68		46.13	46.68	44.98	45.68	45.68	45.68
8		45.68	46.81		45.93	45.68	46.81	46.81	46.81	46.81
9		45.44	48.09		46.26	45.82	45.82	46.76	46.76	46.76
10		68.54	71.09		69.86	69.43	69.32	70.86	70.86	70.86
11		51.61	53.55		51.61	51.61	52.97	53.35	53.35	53.35
12		48.7	52.97		49.67	49.67	50.06	51.03	51.61	51.03
13		44.61	47.85		45.26	45.26	44.93	44.61	44.61	44.61
14		44.86	45.96		44.92	44.92	45.41	45.41	45.41	45.96
15		45.68	49		48.17	48.17	48.03	48.17	48.03	48.31
16		47.05	48.48		47.41	47.41	47.52	48.12	48.12	48.12
17		39.02	42.69		41.75	41.75	41.54	42.48	42.48	42.48
18		47.84	50.25		48.57	48.57	49.1	48.89	49.62	48.78
19		40.03	43.66		43.2	43.2	42.08	42.36	42.55	42.55
20		45.06	47.3		46.46	46.46	45.99	47.02	47.02	46.55
21		50.22	51.62		50.22	50.22	50.22	50.22	50.64	50.64
22		51.76	54.99		54.43	54.43	54.71	52.45	54.71	54.71
23		54.88	58.17		57.23	57.23	56.76	55.82	57.94	57.82
24		54.59	56.7		55.5	55.5	56.5	55.09	55.9	55.9
25		57.11	58.92		57.9	57.9	58.32	57.11	58.61	57.81
26		48.03	51.62		50.48	50.48	51.36	48.21	50.05	50.05
27		50.64	53.26		51.87	51.87	51.41	50.94	52.02	51.87
28		54.72	57.5		57.04	57.04	56.65	55.11	57.35	57.35
29		46.89	47.58		46.89	46.89	46.89	46.89	47.58	47.58
30		51.88	54.23		52.83	52.83	53.24	52.37	53.89	53.76
31		50.29	52.24		51.63	51.63	51.51	51.51	51.63	51.63
32		49.81	52.24		51.99	51.99	50.78	51.14	51.14	51.14
33		51.27	53.1		51.78	51.78	52.7	53.1	51.67	51.67
34		46.45	48.81		47.24	47.24	47.5	48.81	48.81	48.81
35		49.08	52.75		51.18	51.18	51.18	51.35	51.88	51.18
36		47.98	49.35		47.98	47.98	48.43	48.66	48.66	48.43
37		52.71	53.11		52.71	52.71	52.71	52.71	52.71	52.85
38		54.05	56.61		55	55	56.34	56.21	56.34	56.21
39		50.61	52.64		51.8	51.8	52.1	52.4	52.4	52.4
40		55.28	56.6		55.7	55.7	55.28	56.6	56.6	56.6
41		46.27	53.4		51.71	51.71	52	52.32	52.32	52.32
42		49.98	55.03		51.95	51.95	50.95	51.99	51.99	51.99
43		54.56	54.87		54.56	54.56	54.56	54.87	54.87	54.87
44		49.47	52.77		49.47	49.47	51.19	52.5	52.5	52.5
45		39.81	46.87		45.69	45.69	45.69	46.79	46.79	45.4
46		56.28	57.22		56.99	56.99	56.63	56.93	56.93	56.93
47		57.04	57.98		57.45	57.45	57.61	57.56	57.72	57.72
48		58.93	60.36		59.08	59.08	59.88	60.36	60.36	60.36
49		53.8	54.83		53.8	53.8	54.04	54.6	54.6	54.6
50		54.79	54.79		54.79	54.79	54.79	54.79	54.79	54.79
51		55.83	59.94		57.39	57.39	57.07	57.66	59.94	57.66
52		52.58	53.99		53.08	53.08	53.57	53.85	53.85	53.71
53		51.53	52.58		51.53	51.53	51.74	52.58	52.58	52.58
54		50.97	53.27		52	52	51.94	53.27	52.54	52.54
55		51.94	52.72		52.72	52.72	52.18	51.94	51.94	52.3
56		52.73	55.48		54	54	54.15	54.96	54.96	54.96
57		57.48	58.61		58.33	58.33	57.82	58.43	58.43	58.43
58		52.13	55.52		53.49	53.49	53.72	52.88	54.11	54.16
59		52	53.69		52.68	52.68	52.13	52.96	52.77	53.19
60		41.87	49.79		48.08	48.08	48.5	49.71	49.71	49.71

Summary of results after 50 iterations

					Simulated annealing results - Linear cooling	Simulated annealing results - High starting temperature	Simulated annealing results - Standard parameters	Tentacle Algorithm - Accelorated growth	Tentacle Algorithm - Flexible	Tentacle Algorithm - Standard
Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 50 iterations	Max recovery after 50 iterations	Max recovery after 50 iterations	Max recovery after 50 iterations	Max recovery after 50 iterations	Max recovery after 50 iterations
Avg.		49.872	52.378		50.781	50.798	50.877	51.276	51.521	51.503
1		35.86	43.32		39.12	37.26	39.59	38.66	38.66	38.66
2		41.51	43.78		41.51	42.85	42.85	42.44	42.44	42.44
3		50.36	53.66		51.67	52.44	51.81	51.91	51.91	51.91
4		45.7	46.29		46.09	45.7	45.7	46.09	46.09	46.09
5		43.73	44.06		43.73	43.73	43.73	44.06	44.06	44.06
6		48.84	51.41		49.84	50.41	49.55	51.12	51.12	51.12
7		43.93	46.68		46.13	46.68	43.93	45.68	45.68	45.68
8		45.68	46.81		45.93	45.68	46.81	46.68	46.68	46.68
9		45.44	48.09		46.26	45.66	45.44	46.76	46.76	46.76
10		68.54	71.09		68.54	69.43	68.54	70.86	70.86	70.86
11		51.61	53.55		51.61	51.61	51.8	53.35	53.35	53.35
12		48.7	52.97		49.67	49.67	50.06	50.83	50.83	50.83
13		44.61	47.85		45.26	45.26	44.61	44.61	44.61	44.61
14		44.86	45.96		44.92	44.92	45.41	45.41	45.41	45.41
15		45.68	49		48.17	48.17	47.06	47.6	47.6	48.31
16		47.05	48.48		47.41	47.41	47.17	48	48	48
17		39.02	42.69		41.75	41.75	40.76	42.48	42.48	42.48
18		47.84	50.25		48.47	48.47	48.68	48.78	49.62	48.78
19		40.03	43.66		42.36	42.36	41.62	42.36	42.55	42.55
20		45.06	47.3		45.9	45.9	45.99	46.55	46.55	46.55
21		50.22	51.62		50.22	50.22	50.22	50.22	50.64	50.64
22		51.76	54.99		51.76	51.76	54.71	52.45	54.71	54.71
23		54.88	58.17		57.23	57.23	55.23	55.82	57.82	57.82
24		54.59	56.7		55.5	55.5	56.5	55.09	55.9	55.9
25		57.11	58.92		57.11	57.11	58.32	57.11	57.91	57.21
26		48.03	51.62		48.85	48.85	49.52	48.21	50.05	50.05
27		50.64	53.26		51.87	51.87	50.64	50.94	51.87	51.87
28		54.72	57.5		56.42	56.42	56.58	55.11	57.35	57.35
29		46.89	47.58		46.89	46.89	46.89	46.89	47.58	47.58
30		51.88	54.23		52.64	52.64	52.15	52.37	53.69	53.69
31		50.29	52.24		51.26	51.26	51.14	51.51	51.51	51.51
32		49.81	52.24		50.78	50.78	50.78	51.14	51.14	51.14
33		51.27	53.1		51.78	51.78	51.37	51.67	51.67	51.67
34		46.45	48.81		47.24	47.24	46.8	48.81	47.5	47.68
35		49.08	52.75		50.99	50.99	51.18	51.18	51.88	51.18
36		47.98	49.35		47.98	47.98	48.05	48.43	48.43	48.43
37		52.71	53.11		52.71	52.71	52.71	52.71	52.71	52.71
38		54.05	56.61		54.46	54.46	55.26	56.21	56.21	56.21
39		50.61	52.64		51.8	51.8	52.1	52.28	52.28	52.28
40		55.28	56.6		55.28	55.28	55.28	56.6	56.6	56.6
41		46.27	53.4		50.86	50.86	51.65	52.23	52.23	52.23
42		49.98	55.03		51.41	51.41	49.98	51.99	51.99	51.99
43		54.56	54.87		54.56	54.56	54.56	54.87	54.87	54.87
44		49.47	52.77		49.47	49.47	51.19	52.5	52.5	52.5
45		39.81	46.87		41.42	41.42	45.69	45.4	45.4	45.4
46		56.28	57.22		56.28	56.28	56.63	56.93	56.93	56.93
47		57.04	57.98		57.45	57.45	57.3	57.56	57.72	57.72
48		58.93	60.36		59.08	59.08	59.88	60.36	60.36	60.36
49		53.8	54.83		53.8	53.8	54.04	54.6	54.6	54.6
50		54.79	54.79		54.79	54.79	54.79	54.79	54.79	54.79
51		55.83	59.94		55.83	55.83	57.07	57.66	57.66	57.66
52		52.58	53.99		53.08	53.08	53.57	53.71	53.71	53.71
53		51.53	52.58		51.53	51.53	51.53	52.58	52.58	52.58
54		50.97	53.27		52	52	51.94	52.54	52.54	52.54
55		51.94	52.72		51.94	51.94	51.94	51.94	51.94	51.94
56		52.73	55.48		54	54	53.78	54.26	54.96	54.96
57		57.48	58.61		58.33	58.33	57.77	58.33	58.33	58.33
58		52.13	55.52		53.49	53.49	52.13	52.83	52.97	52.83
59		52	53.69		52.34	52.34	52.13	52.77	52.77	53.19
60		41.87	49.79		48.08	48.08	48.5	49.71	49.71	49.71

Appendix C: Results for Simulated Annealing and Tentacle Algorithm after 50, 90 and 350 iterations

Simulated annealing results - Linear cooling

Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 50 iterations	Max recovery after 90 iterations	Max recovery after 350 iterations
Avg.		49.872	52.378		50.781	51.146	51.797
1		35.86	43.32		39.12	39.12	41.68
2		41.51	43.78		41.51	42.91	42.91
3		50.36	53.66		51.67	51.67	51.85
4		45.7	46.29		46.09	46.09	46.09
5		43.73	44.06		43.73	43.73	43.73
6		48.84	51.41		49.84	49.84	51.12
7		43.93	46.68		46.13	46.13	46.13
8		45.68	46.81		45.93	45.93	45.93
9		45.44	48.09		46.26	46.26	47.87
10		68.54	71.09		68.54	69.86	70.42
11		51.61	53.55		51.61	51.61	53.16
12		48.7	52.97		49.67	49.67	51.61
13		44.61	47.85		45.26	45.26	46.79
14		44.86	45.96		44.92	44.92	45.96
15		45.68	49		48.17	48.17	48.31
16		47.05	48.48		47.41	47.41	47.41
17		39.02	42.69		41.75	41.75	42.48
18		47.84	50.25		48.47	48.57	50.25
19		40.03	43.66		42.36	43.2	43.29
20		45.06	47.3		45.9	46.46	46.46
21		50.22	51.62		50.22	50.22	50.22
22		51.76	54.99		51.76	54.43	54.43
23		54.88	58.17		57.23	57.23	57.47
24		54.59	56.7		55.5	55.5	56.2
25		57.11	58.92		57.11	57.9	58.52
26		48.03	51.62		48.85	50.48	50.48
27		50.64	53.26		51.87	51.87	52.33
28		54.72	57.5		56.42	57.04	57.04
29		46.89	47.58		46.89	46.89	47.16
30		51.88	54.23		52.64	52.83	53.76
31		50.29	52.24		51.26	51.63	51.75
32		49.81	52.24		50.78	51.99	51.99
33		51.27	53.1		51.78	51.78	52.49
34		46.45	48.81		47.24	47.24	48.03
35		49.08	52.75		50.99	51.18	51.7
36		47.98	49.35		47.98	47.98	49.35
37		52.71	53.11		52.71	52.71	52.71
38		54.05	56.61		54.46	55	55.6
39		50.61	52.64		51.8	51.8	52.64
40		55.28	56.6		55.28	55.7	55.88
41		46.27	53.4		50.86	51.71	52.84
42		49.98	55.03		51.41	51.95	54.47
43		54.56	54.87		54.56	54.56	54.56
44		49.47	52.77		49.47	49.47	52.5
45		39.81	46.87		41.42	45.69	46.87
46		56.28	57.22		56.28	56.99	57.04
47		57.04	57.98		57.45	57.45	57.98
48		58.93	60.36		59.08	59.08	59.29
49		53.8	54.83		53.8	53.8	54.69
50		54.79	54.79		54.79	54.79	54.79
51		55.83	59.94		55.83	57.39	57.68
52		52.58	53.99		53.08	53.08	53.85
53		51.53	52.58		51.53	51.53	52.58
54		50.97	53.27		52	52	52.9
55		51.94	52.72		51.94	52.72	52.72
56		52.73	55.48		54	54	55.48
57		57.48	58.61		58.33	58.33	58.61
58		52.13	55.52		53.49	53.49	53.49
59		52	53.69		52.34	52.68	52.71
60		41.87	49.79		48.08	48.08	49.58

Simulated annealing results - High starting temperature

Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 50 iterations	Max recovery after 90 iterations	Max recovery after 350 iterations
Avg.		49.872	52.378		50.798	51.132	51.810
1		35.86	43.32		37.26	37.26	40.99
2		41.51	43.78		42.85	42.85	42.85
3		50.36	53.66		52.44	52.44	52.44
4		45.7	46.29		45.7	45.7	45.7
5		43.73	44.06		43.73	43.73	43.73
6		48.84	51.41		50.41	51.12	51.12
7		43.93	46.68		46.68	46.68	46.68
8		45.68	46.81		45.68	45.68	46.68
9		45.44	48.09		45.66	45.82	47.2
10		68.54	71.09		69.43	69.43	71.09
11		51.61	53.55		51.61	51.61	53.16
12		48.7	52.97		49.67	49.67	51.61
13		44.61	47.85		45.26	45.26	46.79
14		44.86	45.96		44.92	44.92	45.96
15		45.68	49		48.17	48.17	48.31
16		47.05	48.48		47.41	47.41	47.41
17		39.02	42.69		41.75	41.75	42.48
18		47.84	50.25		48.47	48.57	50.25
19		40.03	43.66		42.36	43.2	43.29
20		45.06	47.3		45.9	46.46	46.46
21		50.22	51.62		50.22	50.22	50.22
22		51.76	54.99		51.76	54.43	54.43
23		54.88	58.17		57.23	57.23	57.47
24		54.59	56.7		55.5	55.5	56.2
25		57.11	58.92		57.11	57.9	58.52
26		48.03	51.62		48.85	50.48	50.48
27		50.64	53.26		51.87	51.87	52.33
28		54.72	57.5		56.42	57.04	57.04
29		46.89	47.58		46.89	46.89	47.16
30		51.88	54.23		52.64	52.83	53.76
31		50.29	52.24		51.26	51.63	51.75
32		49.81	52.24		50.78	51.99	51.99
33		51.27	53.1		51.78	51.78	52.49
34		46.45	48.81		47.24	47.24	48.03
35		49.08	52.75		50.99	51.18	51.7
36		47.98	49.35		47.98	47.98	49.35
37		52.71	53.11		52.71	52.71	52.71
38		54.05	56.61		54.46	55	55.6
39		50.61	52.64		51.8	51.8	52.64
40		55.28	56.6		55.28	55.7	55.88
41		46.27	53.4		50.86	51.71	52.84
42		49.98	55.03		51.41	51.95	54.47
43		54.56	54.87		54.56	54.56	54.56
44		49.47	52.77		49.47	49.47	52.5
45		39.81	46.87		41.42	45.69	46.87
46		56.28	57.22		56.28	56.99	57.04
47		57.04	57.98		57.45	57.45	57.98
48		58.93	60.36		59.08	59.08	59.29
49		53.8	54.83		53.8	53.8	54.69
50		54.79	54.79		54.79	54.79	54.79
51		55.83	59.94		55.83	57.39	57.68
52		52.58	53.99		53.08	53.08	53.85
53		51.53	52.58		51.53	51.53	52.58
54		50.97	53.27		52	52	52.9
55		51.94	52.72		51.94	52.72	52.72
56		52.73	55.48		54	54	55.48
57		57.48	58.61		58.33	58.33	58.61
58		52.13	55.52		53.49	53.49	53.49
59		52	53.69		52.34	52.68	52.71
60		41.87	49.79		48.08	48.08	49.58

Simulated annealing results - Standard parameters

Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 50 iterations	Max recovery after 90 iterations	Max recovery after 350 iterations
Avg.		49.872	52.378		50.877	51.238	51.818
1		35.86	43.32		39.59	40.05	41.21
2		41.51	43.78		42.85	42.85	43.32
3		50.36	53.66		51.81	51.81	53.47
4		45.7	46.29		45.7	45.7	46.29
5		43.73	44.06		43.73	43.73	44.06
6		48.84	51.41		49.55	50.84	50.84
7		43.93	46.68		43.93	44.98	45.13
8		45.68	46.81		46.81	46.81	46.81
9		45.44	48.09		45.44	45.82	47.32
10		68.54	71.09		68.54	69.32	71.09
11		51.61	53.55		51.8	52.97	52.97
12		48.7	52.97		50.06	50.06	51.22
13		44.61	47.85		44.61	44.93	45.26
14		44.86	45.96		45.41	45.41	45.41
15		45.68	49		47.06	48.03	48.72
16		47.05	48.48		47.17	47.52	48
17		39.02	42.69		40.76	41.54	42.69
18		47.84	50.25		48.68	49.1	49.1
19		40.03	43.66		41.62	42.08	43.48
20		45.06	47.3		45.99	45.99	45.99
21		50.22	51.62		50.22	50.22	50.78
22		51.76	54.99		54.71	54.71	54.71
23		54.88	58.17		55.23	56.76	57.82
24		54.59	56.7		56.5	56.5	56.5
25		57.11	58.92		58.32	58.32	58.52
26		48.03	51.62		49.52	51.36	51.62
27		50.64	53.26		50.64	51.41	51.87
28		54.72	57.5		56.58	56.65	56.81
29		46.89	47.58		46.89	46.89	47.58
30		51.88	54.23		52.15	53.24	54.23
31		50.29	52.24		51.14	51.51	51.87
32		49.81	52.24		50.78	50.78	51.51
33		51.27	53.1		51.37	52.7	53.1
34		46.45	48.81		46.8	47.5	48.81
35		49.08	52.75		51.18	51.18	51.88
36		47.98	49.35		48.05	48.43	48.43
37		52.71	53.11		52.71	52.71	52.98
38		54.05	56.61		55.26	56.34	56.34
39		50.61	52.64		52.1	52.1	52.28
40		55.28	56.6		55.28	55.28	56.12
41		46.27	53.4		51.65	52	53.4
42		49.98	55.03		49.98	50.95	52
43		54.56	54.87		54.56	54.56	54.56
44		49.47	52.77		51.19	51.19	52.63
45		39.81	46.87		45.69	45.69	45.69
46		56.28	57.22		56.63	56.63	57.16
47		57.04	57.98		57.3	57.61	57.66
48		58.93	60.36		59.88	59.88	60.36
49		53.8	54.83		54.04	54.04	54.6
50		54.79	54.79		54.79	54.79	54.79
51		55.83	59.94		57.07	57.07	59.44
52		52.58	53.99		53.57	53.57	53.85
53		51.53	52.58		51.53	51.74	52.16
54		50.97	53.27		51.94	51.94	53.27
55		51.94	52.72		51.94	52.18	52.3
56		52.73	55.48		53.78	54.15	54.43
57		57.48	58.61		57.77	57.82	58.43
58		52.13	55.52		52.13	53.72	54.15
59		52	53.69		52.13	52.13	52.71
60		41.87	49.79		48.5	48.5	49.33

Tentacle Algorithm - Accelorated growth

Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 50 iterations	Max recovery after 90 iterations	Max recovery after 350 iterations		Tentacle length at 150 iterations	Tentacle length at end
Avg.		49.872	52.378		51.276	51.425	51.568		4.333	14.250
1		35.86	43.32		38.66	40.29	40.29		5	18
2		41.51	43.78		42.44	42.85	42.85		5	18
3		50.36	53.66		51.91	51.91	52.3		7	12
4		45.7	46.29		46.09	46.09	46.09		7	18
5		43.73	44.06		44.06	44.06	44.06		7	18
6		48.84	51.41		51.12	51.12	51.12		9	18
7		43.93	46.68		45.68	45.68	46.68		7	5
8		45.68	46.81		46.68	46.81	46.81		2	18
9		45.44	48.09		46.76	46.76	46.76		7	18
10		68.54	71.09		70.86	70.86	70.86		5	18
11		51.61	53.55		53.35	53.35	53.35		9	18
12		48.7	52.97		50.83	51.03	51.03		3	18
13		44.61	47.85		44.61	44.61	46.55		9	12
14		44.86	45.96		45.41	45.41	45.41		7	18
15		45.68	49		47.6	48.17	48.17		2	18
16		47.05	48.48		48	48.12	48.12		5	18
17		39.02	42.69		42.48	42.48	42.48		7	18
18		47.84	50.25		48.78	48.89	50.25		2	7
19		40.03	43.66		42.36	42.36	42.36		5	18
20		45.06	47.3		46.55	47.02	47.11		1	18
21		50.22	51.62		50.22	50.22	50.22		0	0
22		51.76	54.99		52.45	52.45	52.45		0	0
23		54.88	58.17		55.82	55.82	55.82		0	0
24		54.59	56.7		55.09	55.09	55.09		0	0
25		57.11	58.92		57.11	57.11	57.11		0	0
26		48.03	51.62		48.21	48.21	48.21		0	0
27		50.64	53.26		50.94	50.94	50.94		0	0
28		54.72	57.5		55.11	55.11	55.11		0	0
29		46.89	47.58		46.89	46.89	46.89		0	0
30		51.88	54.23		52.37	52.37	52.37		0	0
31		50.29	52.24		51.51	51.51	51.51		5	18
32		49.81	52.24		51.14	51.14	51.14		7	18
33		51.27	53.1		51.67	53.1	53.1		3	18
34		46.45	48.81		48.81	48.81	48.81		5	18
35		49.08	52.75		51.18	51.35	51.35		3	18
36		47.98	49.35		48.43	48.66	49.19		2	18
37		52.71	53.11		52.71	52.71	52.71		9	18
38		54.05	56.61		56.21	56.21	56.21		7	18
39		50.61	52.64		52.28	52.4	52.4		3	18
40		55.28	56.6		56.6	56.6	56.6		5	18
41		46.27	53.4		52.23	52.32	52.32		3	18
42		49.98	55.03		51.99	51.99	51.99		7	18
43		54.56	54.87		54.87	54.87	54.87		7	18
44		49.47	52.77		52.5	52.5	52.5		7	18
45		39.81	46.87		45.4	46.79	46.79		3	18
46		56.28	57.22		56.93	56.93	56.99		5	15
47		57.04	57.98		57.56	57.56	57.72		5	15
48		58.93	60.36		60.36	60.36	60.36		7	18
49		53.8	54.83		54.6	54.6	54.6		7	18
50		54.79	54.79		54.79	54.79	54.79		9	18
51		55.83	59.94		57.66	57.66	57.66		9	18
52		52.58	53.99		53.71	53.85	53.85		3	18
53		51.53	52.58		52.58	52.58	52.58		7	18
54		50.97	53.27		52.54	53.27	53.27		3	18
55		51.94	52.72		51.94	51.94	52.3		2	18
56		52.73	55.48		54.26	54.96	54.96		3	18
57		57.48	58.61		58.33	58.43	58.43		3	18
58		52.13	55.52		52.83	52.88	55.52		1	15
59		52	53.69		52.77	52.96	52.96		2	18
60		41.87	49.79		49.71	49.71	49.71		7	18

Tentacle Algorithm - Flexible

Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 50 iterations	Max recovery after 90 iterations	Max recovery after 350 iterations		Tentacle length at 150 iterations	Tentacle length at end
		49.872	52.378		51.521	51.705	51.893		3.800	7.333
1		35.86	43.32		38.66	38.66	42.38		6	5
2		41.51	43.78		42.44	43.32	43.32		1	9
3		50.36	53.66		51.91	51.91	51.91		5	9
4		45.7	46.29		46.09	46.09	46.09		2	9
5		43.73	44.06		44.06	44.06	44.06		5	9
6		48.84	51.41		51.12	51.12	51.12		6	9
7		43.93	46.68		45.68	45.68	45.88		5	9
8		45.68	46.81		46.68	46.81	46.81		5	6
9		45.44	48.09		46.76	46.76	46.76		2	8
10		68.54	71.09		70.86	70.86	70.86		4	5
11		51.61	53.55		53.35	53.35	53.35		6	9
12		48.7	52.97		50.83	51.61	51.61		2	1
13		44.61	47.85		44.61	44.61	44.61		3	9
14		44.86	45.96		45.41	45.41	45.41		5	9
15		45.68	49		47.6	48.03	48.31		2	9
16		47.05	48.48		48	48.12	48.12		1	8
17		39.02	42.69		42.48	42.48	42.48		5	9
18		47.84	50.25		49.62	49.62	49.62		3	2
19		40.03	43.66		42.55	42.55	42.83		2	7
20		45.06	47.3		46.55	47.02	47.2		5	7
21		50.22	51.62		50.64	50.64	50.64		5	9
22		51.76	54.99		54.71	54.71	54.99		4	1
23		54.88	58.17		57.82	57.94	58.17		3	9
24		54.59	56.7		55.9	55.9	56.7		5	1
25		57.11	58.92		57.91	58.61	58.81		3	9
26		48.03	51.62		50.05	50.05	51.62		5	4
27		50.64	53.26		51.87	52.02	52.02		5	9
28		54.72	57.5		57.35	57.35	57.35		3	9
29		46.89	47.58		47.58	47.58	47.58		4	9
30		51.88	54.23		53.69	53.89	54.17		5	9
31		50.29	52.24		51.51	51.63	51.63		3	4
32		49.81	52.24		51.14	51.14	51.26		2	6
33		51.27	53.1		51.67	51.67	52.08		1	8
34		46.45	48.81		47.5	48.81	48.81		4	9
35		49.08	52.75		51.88	51.88	51.88		3	9
36		47.98	49.35		48.43	48.66	49.19		2	9
37		52.71	53.11		52.71	52.71	53.11		6	9
38		54.05	56.61		56.21	56.34	56.34		5	3
39		50.61	52.64		52.28	52.4	52.4		3	9
40		55.28	56.6		56.6	56.6	56.6		4	9
41		46.27	53.4		52.23	52.32	52.32		1	8
42		49.98	55.03		51.99	51.99	52		5	6
43		54.56	54.87		54.87	54.87	54.87		5	9
44		49.47	52.77		52.5	52.5	52.5		5	9
45		39.81	46.87		45.4	46.79	46.79		4	2
46		56.28	57.22		56.93	56.93	56.93		4	3
47		57.04	57.98		57.72	57.72	57.72		4	3
48		58.93	60.36		60.36	60.36	60.36		5	9
49		53.8	54.83		54.6	54.6	54.6		2	9
50		54.79	54.79		54.79	54.79	54.79		6	9
51		55.83	59.94		57.66	59.94	59.94		6	9
52		52.58	53.99		53.71	53.85	53.85		1	8
53		51.53	52.58		52.58	52.58	52.58		5	9
54		50.97	53.27		52.54	52.54	52.54		5	9
55		51.94	52.72		51.94	51.94	52.3		6	4
56		52.73	55.48		54.96	54.96	54.96		1	8
57		57.48	58.61		58.33	58.43	58.43		3	9
58		52.13	55.52		52.97	54.11	55.52		2	7
59		52	53.69		52.77	52.77	52.77		3	9
60		41.87	49.79		49.71	49.71	49.71		5	9

Tentacle Algorithm - Standard

Block number		Horns up recovery	Maximum recovery (Exhaustive search)		Max recovery after 50 iterations	Max recovery after 90 iterations	Max recovery after 350 iterations		Tentacle length at 150 iterations	Tentacle length at end
Avg.		49.872	52.378		51.503	51.622	51.897		4.017	8.000
1		35.86	43.32		38.66	40.29	40.29		4	9
2		41.51	43.78		42.44	42.85	42.85		4	9
3		50.36	53.66		51.91	51.91	51.91		5	9
4		45.7	46.29		46.09	46.09	46.09		5	9
5		43.73	44.06		44.06	44.06	44.06		5	9
6		48.84	51.41		51.12	51.12	51.12		6	9
7		43.93	46.68		45.68	45.68	46.68		5	7
8		45.68	46.81		46.68	46.81	46.81		2	9
9		45.44	48.09		46.76	46.76	46.76		5	9
10		68.54	71.09		70.86	70.86	70.86		4	9
11		51.61	53.55		53.35	53.35	53.35		6	9
12		48.7	52.97		50.83	51.03	51.03		3	9
13		44.61	47.85		44.61	44.61	47.2		6	6
14		44.86	45.96		45.41	45.96	45.96		5	9
15		45.68	49		48.31	48.31	48.72		4	5
16		47.05	48.48		48	48.12	48.12		4	9
17		39.02	42.69		42.48	42.48	42.48		5	9
18		47.84	50.25		48.78	48.78	49.94		2	6
19		40.03	43.66		42.55	42.55	43.66		2	7
20		45.06	47.3		46.55	46.55	47.2		2	7
21		50.22	51.62		50.64	50.64	50.64		5	9
22		51.76	54.99		54.71	54.71	54.71		4	9
23		54.88	58.17		57.82	57.82	57.94		5	9
24		54.59	56.7		55.9	55.9	56.7		5	7
25		57.11	58.92		57.21	57.81	58.92		2	9
26		48.03	51.62		50.05	50.05	50.74		5	7
27		50.64	53.26		51.87	51.87	52.02		1	5
28		54.72	57.5		57.35	57.35	57.35		6	9
29		46.89	47.58		47.58	47.58	47.58		4	9
30		51.88	54.23		53.69	53.76	53.76		3	3
31		50.29	52.24		51.51	51.63	51.75		2	9
32		49.81	52.24		51.14	51.14	51.14		5	4
33		51.27	53.1		51.67	51.67	53.1		2	9
34		46.45	48.81		47.68	48.81	48.81		4	9
35		49.08	52.75		51.18	51.18	51.53		1	9
36		47.98	49.35		48.43	48.43	49.19		5	5
37		52.71	53.11		52.71	52.85	53.11		6	9
38		54.05	56.61		56.21	56.21	56.54		5	9
39		50.61	52.64		52.28	52.4	52.4		3	9
40		55.28	56.6		56.6	56.6	56.6		4	9
41		46.27	53.4		52.23	52.32	52.32		2	9
42		49.98	55.03		51.99	51.99	52		5	6
43		54.56	54.87		54.87	54.87	54.87		5	9
44		49.47	52.77		52.5	52.5	52.5		5	9
45		39.81	46.87		45.4	45.4	46.72		3	9
46		56.28	57.22		56.93	56.93	57.16		4	7
47		57.04	57.98		57.72	57.72	57.72		2	9
48		58.93	60.36		60.36	60.36	60.36		5	9
49		53.8	54.83		54.6	54.6	54.6		3	9
50		54.79	54.79		54.79	54.79	54.79		6	9
51		55.83	59.94		57.66	57.66	57.66		4	9
52		52.58	53.99		53.71	53.71	53.85		4	9
53		51.53	52.58		52.58	52.58	52.58		5	9
54		50.97	53.27		52.54	52.54	52.54		5	9
55		51.94	52.72		51.94	52.3	52.72		6	6
56		52.73	55.48		54.96	54.96	54.96		2	9
57		57.48	58.61		58.33	58.43	58.43		3	9
58		52.13	55.52		52.83	54.16	55.52		2	4
59		52	53.69		53.19	53.19	53.19		4	1
60		41.87	49.79		49.71	49.71	49.71		5	9

Appendix D: Additional description of methodology of Simulated Annealing utilization

Project : LC19_1	Run performed on 28-Nov-03 at 12.45.50pm									
LOG I	L190360/02/06025117									
Log. No	Diamete	Length	Taper (r	Sweep	Ovality	Sawing	Offset	Alignme	Rotatior	Recovery
1	18.2	3.6	5.5	23.2	0.87	1	-30	-30	0	30.94
1	18.2	3.6	5.5	23.2	0.87	1	-25	-30	0	29.37
1	18.2	3.6	5.5	23.2	0.87	1	-20	-30	0	29.17
1	18.2	3.6	5.5	23.2	0.87	1	-15	-30	0	33.79
1	18.2	3.6	5.5	23.2	0.87	1	-10	-30	0	36.07
1	18.2	3.6	5.5	23.2	0.87	1	-5	-30	0	34.43
1	18.2	3.6	5.5	23.2	0.87	1	0	-30	0	35.58
1	18.2	3.6	5.5	23.2	0.87	1	5	-30	0	31.56
1	18.2	3.6	5.5	23.2	0.87	1	10	-30	0	31.03
1	18.2	3.6	5.5	23.2	0.87	1	15	-30	0	34.76
1										30.98
1										29.23
1										27.14
1										28.18
1										28.78
1										28.24
1										32.37
1										35.6
1										35.13
1										34.89
1	18.2	3.6	5.5	23.2	0.87	1	5	-25	0	34.63
1	18.2	3.6	5.5	23.2	0.87	1	10	-25	0	29.64
1	18.2	3.6	5.5	23.2	0.87	1	15	-25	0	33.13
1	18.2	3.6	5.5	23.2	0.87	1	20	-25	0	34.76
1	18.2	3.6	5.5	23.2	0.87	1	25	-25	0	29.23
1	18.2	3.6	5.5	23.2	0.87	1	30	-25	0	29.7
1	18.2	3.6	5.5	23.2	0.87	1	-30	-20	0	30.23
1	18.2	3.6	5.5	23.2	0.87	1	-25	-20	0	29.24
1	18.2	3.6	5.5	23.2	0.87	1	-20	-20	0	31.5
1	18.2	3.6	5.5	23.2	0.87	1	-15	-20	0	30.27
1	18.2	3.6	5.5	23.2	0.87	1	-10	-20	0	34.43
1	18.2	3.6	5.5	23.2	0.87	1	-5	-20	0	36.79

1. Data

This section contains the data imported from the SIMSAW simulation package. It contains the volume recovery for al the positional combinations of one log. Only a small section is shown per illustration.

Report	
Horns up starting position	%
Block	1
Exhaustive search	
Horns up recovery	35.86
Maximum recovery	43.32
Simulated Annealing Heuristic	
Recovery after 350 iterations	40.52
Maximum value within above	40.52
Percentage improvement (at 350)	
from horns up to optimum	62.466
Percentage improvement (at max)	
from horns up to optimum	62.466

2. Report

The above shaded area contains the results for the heuristic after a limited number of iterations has been performed.

			ID	Recovery						ID (Part	Offset	Skew	Rotation
10	10	10	101010	30.94			0	0	0	10	-30	-30	0
10	10	11	101011	29.37			0.0769	0.0417	1	11	-25	-25	15
10	10	12	101012	29.17			0.1538	0.0833	2	12	-20	-20	30
10	10	13	101013	33.79			0.2308	0.125	3	13	-15	-15	45
10	10	14	101014	36.07			0.3077	0.1667	4	14	-10	-10	60
10	10	15	101015	34.43			0.3846	0.2083	5	15	-5	-5	75
10	10	16	101016	35.58			0.4615	0.25	6	16	0	0	90
10	10	17	101017	31.56			0.5385	0.2917	7	17	5	5	105
10	10	18	10							18	10	10	120
10	10	19	10							19	15	15	135
10	10	20	10							20	20	20	150
10	10	21	10							21	25	25	165
10	10	22	10							22	30	30	180
10	11	10	10							23			195
10	11	11	10							24			210
10	11	12	10							25			225
10	11	13	10							26			240
10	11	14	10							27			255
10	11	15	10							28			270
10	11	16	10							29			285
10	11	17	10							30			300
10	11	18	10							31			315
10	11	19	10							32			330
10	11	20	101120	34.76				0.9583	23	33			345
10	11	21	101121	29.23				1	24				
10	11	22	101122	29.7									
10	12	10	101210	30.23									

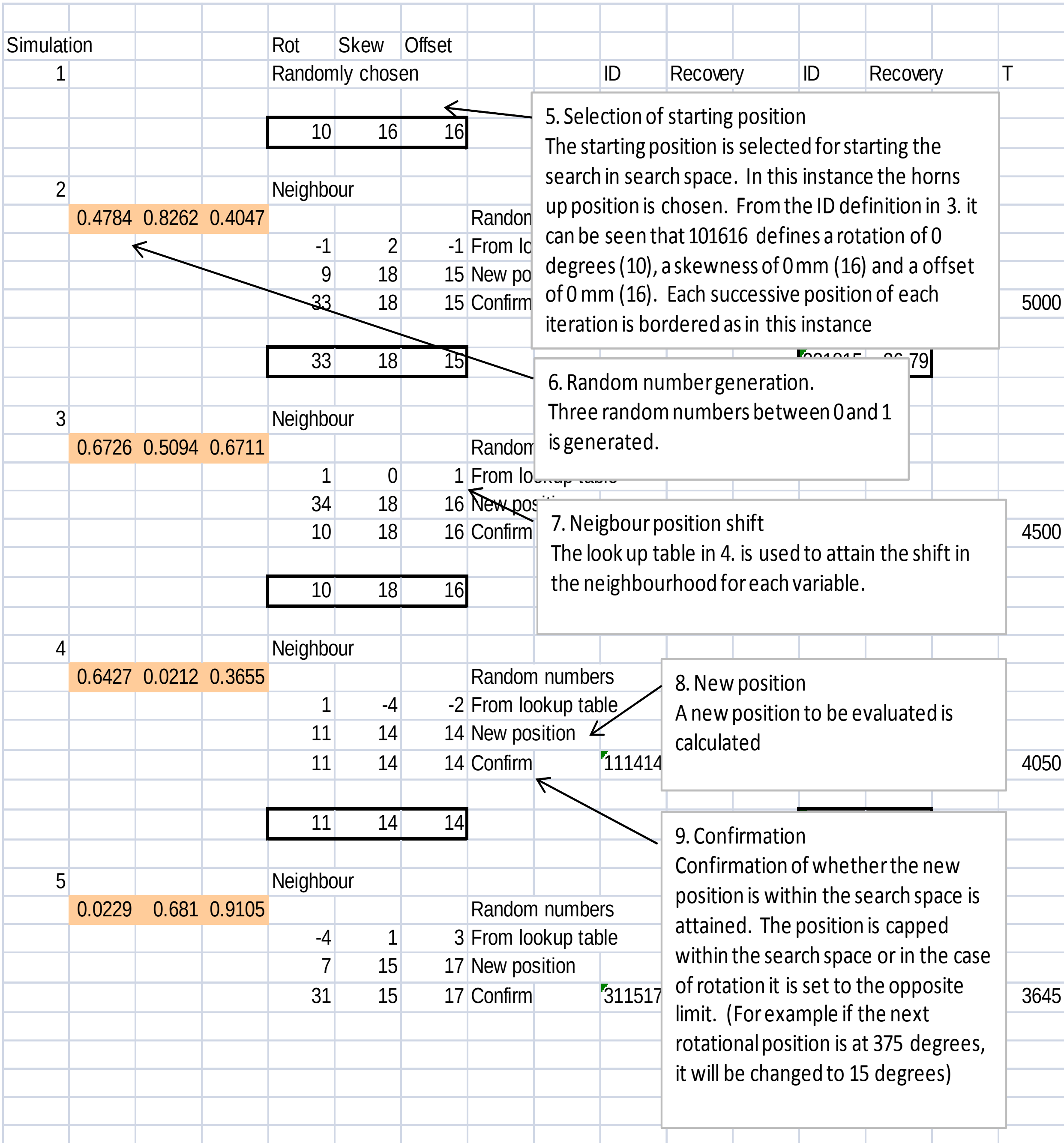
3. Identification of combinations

In these columns the researcher gave unique 6 digit numbers for each combination of positions. The first two digits identifies the rotation, the next two the skewness and the last two the offset. Each identification number (ID) is shown with its recovery.

Neigbour lookup table		
	0	-4
	0.125	-3
	0.25	-2
	0.375	-1
	0.5	0
	0.625	1
	0.75	2
	0.875	3
	1	4

4. Neighbour lookup table:

In the above table the magnitude and direction of the jump to a neighbouring position in the search space is selected (Right hand column). This is based on a random number to be selected between 0 and 1 (Left hand column).


Simulation
Rot Skew Offset
1 Randomly chosen
ID Recovery ID Recovery T
10 16 16
5. Selection of starting position
The starting position is selected for starting the search in search space. In this instance the horns up position is chosen. From the ID definition in 3. it can be seen that 101616 defines a rotation of 0 degrees (10), a skewness of 0 mm (16) and a offset of 0 mm (16). Each successive position of each iteration is bordered as in this instance
2 Neighbour
0.4784 0.8262 0.4047
-1 2 -1
9 18 15
33 18 15
5000
33 18 15
6. Random number generation.
Three random numbers between 0 and 1 is generated.
3 Neighbour
0.6726 0.5094 0.6711
1 0 1
34 18 16
10 18 16
4500
10 18 16
7. Neigbour position shift
The look up table in 4. is used to attain the shift in the neighbourhood for each variable.
4 Neighbour
0.6427 0.0212 0.3655
Random numbers
1 -4 -2 From lookup table
11 14 14 New position
11 14 14 Confirm
111414
4050
11 14 14
8. New position
A new position to be evaluated is calculated
9. Confirmation
Confirmation of whether the new position is within the search space is attained. The position is capped within the search space or in the case of rotation it is set to the opposite limit. (For example if the next rotational position is at 375 degrees, it will be changed to 15 degrees)
5 Neighbour
0.0229 0.681 0.9105
Random numbers
-4 1 3 From lookup table
7 15 17 New position
31 15 17 Confirm
311517
3645

ID	Recovery	ID	Recovery	T	Change	Improved	AF	RN	Forced move

10. New position evaluated
The recovery for the new position is looked up

ID	Recovery	ID	Recovery	T	Change	Improved	AF	RN	Forced move
rs									
le									
301713	30.27			3280.5	-2.68	FALSE	0.9992	0.9007	TRUE
		301713	30.27						

11. Temperature
The temperature is calculated as per cooling schedule. In this instance cooling of 10% takes place every iteration.

ID	Recovery	ID	Recovery	T	Change	Improved	AF	RN	Forced move
rs									
le									
271911	28.07			2952.5	-2.2	FALSE	0.9993	0.302	TRUE
		271911	28.07						
rs									
le									
291610	30.94			2657.2	2.87	TRUE	1.0011	0.434	TRUE
		291610	30.94						
rs									
le									
281510									

12. Decision
The new position's recovery is compared with the old one. Two Boolean operators are generated. The first states whether an improvement in recovery did occur. The second states whether a random number between 0 and 1 is smaller than the exponent of (new recovery minus the old recovery) divided by the Temperature. (See the Simulated annealing Algorithm description) If either of the operators are 'TRUE' the focus will shift to the new position.

The process is repeated until the iteration limit is reached.

ID	Recovery	ID	Recovery	T	Change	Improved	AF	RN	Forced move
rs									
le									
301412	29.56			2152.3	4.29	TRUE	1.002	0.0361	TRUE
		301412	29.56						