A Framework for Search Heuristics

Ross James
Department of Management
University of Canterbury
New Zealand
r.james@mang.canterbury.ac.nz

Abstract

Search heuristics, such as Tabu Search and Simulated Annealing, start from a single solution and incrementally change it in order to find better solutions. Given a particular problem, there are a large number of possible ways of formulating a search in order to solve it. The choice of search space, neighbourhood scheme, search heuristic and its many possible additions and enhancements are all elements of a search formulation. This paper identifies some of the key relationships between these different elements and how these could impact the overall performance of the search. The framework developed from these relationships helps to classify different search heuristic modifications and identifies areas where relatively little research has been done.

1. Introduction

The performance of search heuristics, such as Tabu Search (TS) [3] or Simulated Annealing (SA) [11], can be influenced greatly by the way in which they are formulated. This formulation involves determining the problem representation, which we will refer to as search space design, and the method for changing solutions in order to improve their quality, which we call neighbourhood scheme design. The importance of the formulation on the ability of specific techniques to solve problems is well known in the operations research literature. For example Nemhauser and Wolsey [13] state "In integer programming, formulating a 'good' model is of critical importance to solving the model."

In the search heuristic literature there has been little in the way of detailed consideration of the effect of search space design and neighbourhood scheme design on search performance. Most papers detailing their implementation of a search heuristic say little more than what search space representation and neighbourhood scheme was used. Usually little, if anything, is written about the alternative search space and neighbourhood scheme formulations and the advantages and/or disadvantages of each of these formulations. Often when alternatives are considered all that is said in the papers is something like "Based on our preliminary experiments on neighbourhood structures, insert moves where pairs of jobs are selected are selected at random performed best and we use those moves in our further testing" [10]. No insight is given as to why one scheme performed better or worse than another. The preliminary experiments provide a conclusion but no justification. Authors may have an appropriate justification for their conclusion but omit reporting the details of these. This non-reporting may be a symptom of the phenomenon Hooker [5] identified where only success stories are reported rather than a detailed account of the experimentation process which may lead to other important insights in terms of search heuristic design. The paper by Jain *et al*

[6] is a refreshing exception to this trend and demonstrates the insight that can be gained from in-depth analysis of neighbourhood schemes in particular.

So why does search space and the neighbourhood scheme design affect the performance of a search heuristic?

Firstly the topology of the search space will influence how difficult it is for the search heuristic to find good solutions. A search space with an undulating topology, i.e. with many local optima, relies heavily on the search heuristic's ability to climb out of these local optima for progress to be made. It also means that the search spends a high proportion of its time climbing out of optima rather than improving on the current solution. On the other hand a search where the search space is very smooth is equally very difficult to search, as there is little or no discrimination between neighbouring moves. In this situation the search may find it difficult to determine an appropriate direction to take and could therefore spend a large amount of time making little progress. A search space surface where all solutions lead to the global optima, in what could be called a bowl shaped topology, is the easiest to search as all improving solutions are moving the search towards the global optima.

Secondly the search space determines the number of potential solutions being considered by the search heuristic and therefore defines the enormity of the task being undertaken. Akin to determining the size of the haystack in which we a trying to find a needle. We make a distinction here between the search space and the solution space. We define the solution space as being the set of all possible solutions, both feasible and infeasible. The search space, however, does not have to contain all solutions and nor does it have to be a set of all feasible solutions, but may simply contain data that can be transformed into a solution at a later stage by an algorithm or heuristic. In doing so the search space definition could eliminate infeasible solutions from consideration or only consider solutions that adhere to the known optimality conditions of the problem. When potential solutions are eliminated from the search it is critical that the search does not eliminate optimal solutions to the problem or potentially interesting solutions [15]. It is also important in the design of search spaces that the elimination of potential solutions, even though they may be known to be non-optimal or even infeasible, may have an adverse affect on the search space topology, which may in turn make it more difficult to search. There is therefore a balancing act between reducing the search space size and the likely impact this has on the search space topology.

Thirdly the neighbourhood scheme and any search heuristic specific technique that limits the number of neighbouring solutions being considered, such as candidate lists and random probability distributions, affect the search space topology. These influence the number of possible solutions that the search examines from any one solution in the search space. If there is a large number of neighbouring solutions from a particular point then this will tend to allow a smoother route to better solutions than a scheme that allows only a few neighbouring solutions. However the number of neighbours is not the only consideration. Evans [2] in his comparison of four different neighbourhood schemes for the mean tardiness problem concluded that the performance of the search heuristic was determined more by the coverage of the solution space by the neighbourhood scheme provided rather than just the number of different neighbours defined by neighbourhood scheme.

Along with these design issues we also need to consider the fact that as we get more ideal search space surfaces then there may be a time penalty that needs to be paid. For example, if we defined our neighbourhood scheme to be that ever possible solution was the neighbour to every other possible solutions then we would have the ideal solution space surface, with the global optimal being only one step away from any solution in the solution space. However the time taken to find the correct neighbour to move to is prohibitive in most circumstances and therefore this formulation is not practical. We therefore need to balance the need for the most appropriate design and the practical requirements of an appropriate solution time.

2. Related Literature

The literature that has looked at search space development and design is quite sparse. Storer *et al.* [16] examine the relationship:

Problem Data + Algorithm or Heuristic = Solution

By exploiting the explicit relationship they created two new types of search heuristics which are called “Problem Space” and “Heuristic Space” search heuristics. The Problem Space search changes the problem data but uses the same algorithm or heuristic on the changed data in order to generate a different solution. The Heuristic Space search keeps the problem data constant but changes the heuristic in some manner in order to change the solution being examined. Both of these techniques were tested on the job shop scheduling problem, and compared to the shifting bottleneck heuristic. Results showed that the problem space heuristics developed produced very competitive results and that a combined problem space and heuristic space approach only produced slightly better results than the problem space heuristc by itself.

Tsubakitani and Evans [17] explored collapsing the solution space of the travelling salesman problem (TSP) into a two-dimensional space and then using the two dimensional space as the search space. In their paper they identified there being a mapping between the TSP tour and the location on the (x,y) plane, which in this case represents the search space, and a reverse mapping between the (x,y) plane and the tour, i.e. between the search space and the resulting solution. The mapping between the tour and the cost of the tour was also explicitly identified in their model. The main advantage of this approach was that it allowed the search space to be visualised by mapping it in two dimensions. Results from 50 eight-city TSP problems showed that a search easily got trapped in local minima and it took a lot of time for the search to find the global minima. Illustrating the fact that solution spaces and neighbourhood schemes that result in almost arbitrary relationships need to rely heavily on the search being able to get out of local minima in order to perform well.

James and Buchanan [8] used the concept of a search space mapping to solutions via a heuristic. The search space was represented by a vector of binary variables which was then used by a heuristic to produce a schedule for the early/tardy scheduling problem. Their experiments showed that designing a search space using a heuristic which exploits known problem characteristics can quickly move the search into good areas of the solutions space, however a simpler neighbourhood design is better to do the fine tuning of the solution.

Vaessens *et al.* [18] proposed a template for local search in order to generalise the process of local search and to identify opportunities for development in the search heuristic literature. In their template they use functions called “Generate Neighbours”, which generates all possible neighbours that can be considered by the search at a particular generation, and “Reduce Neighbours”, which reduces the potential neighbours that can be considered. In this paper we are primarily concerned with these two functions as they influence the search space topology. Along with the design of the functions that make up the search which Vaessens *et al.* [18] deals with, in this paper we are also deal with the actual design of the search space as well.

3. Framework Development

We now want to examine the elements that influence the search space topology and the relationships between these elements.

The first element in the framework is the way the problem is represented, what we have been referring to as the search space design. The same problem could be represented in many different ways. For example the problem representation could be a sequence of unique numbers, a sequence of 0/1 variables, a sequence of real numbers, a single number or a set of numbers. How the problem is represented will dictate how many of the remaining elements work and how efficiently they work.

The neighbourhood scheme is strongly related to the way the problem is represented and therefore the design of the search space. Often search space design is carried out before the neighbourhood scheme designed, however since these two elements are very interdependent it is useful to consider the possibilities of each simultaneously as it is the combination of these elements that helps determine the overall effectiveness of the search. For example the choice of the binary search space representation used by James [7] for the early/tardy scheduling problem was inspired by the fact that jobs get classified as either early or tardy. Therefore a way of moving from one potential neighbour to another is to simply change a job's early/tardy classification, and therefore the neighbourhood scheme inspired the 0/1 search space design.

The problem representation also directly affects what an element or instance of the search space actually contains. If it is a direct representation of the problem, then little or no decoding will be required, however other representations may require extensive decoding in order to turn the problem instance into an actual solution to the problem. When this is required an appropriate algorithm or heuristic can be used to convert this instance data into an actual solution to the problem. To do this, the algorithm or heuristic may require some information on the problem in order to convert the instance data into a solution. This will depend extensively on the problem representation. For example, a TSP problem with a sequence of cities problem representation no extra problem data would be required to generate the final tour of the cities. However for an algorithm solving a scheduling problem which uses a sequence of jobs representation, the problem data would be used to determine the times when jobs were started and completed. Information of job processing times is required. The problem data used by the algorithm or heuristic may not necessarily be the original problem data, but could be modified during the search in order to change the way the search operates.

Once the solution is determined by the algorithm or heuristic it can then be evaluated by the objective function so that its quality can be assessed. The objective function will use the original problem data in order to evaluate the quality of the solution. The objective function used does not necessarily have to be the original objective function for the problem. Various changes can be made to the objective function by the search in order to make the different areas of the search space more or less attractive to the search overall.

An important design relationship exists between the objective function and the neighbourhood scheme. The interaction of the objective function and the neighbourhood scheme is critical to the success of the search. The neighbourhood scheme should be designed to try to preserve the majority of the elements of the current solution so that the changes in the objective function value are not too large and therefore adversely affecting the topology of the search space. An example of this is in TSP formulations if you compare a 2-opt neighbourhood scheme with a neighbourhood scheme that swaps the order of two cities in the tour. One reason the 2-opt performs better in this situation is that it only breaks and introduces two arcs in the tour whereas the swap breaks and introduces four arcs. As the arcs are directly related to the objective function, as it is the arcs that determine the cost of the tour. The smaller perturbations of the solution, with respect to the arcs in the solution, mean less variance in the neighbouring solution's objective function values and hence a smoother search space topology.

The actual neighbours to a solution are normally a subset of all the possible neighbours generated from the neighbourhood scheme. Restricting the number of candidate neighbours and/or moving before evaluating all possible neighbours are evaluated can increase the unevenness of the search space topology. This is because the move selected may not be the best possible move in that situation and therefore the variance in objective function value will be higher than what it could have been if all neighbours have be examined. Despite this draw back the use of candidate neighbourhood schemes that restrict the number of neighbours should be considered as many studies ([1], [14], [9]) have shown that the speed gains for a search, such as tabu search, outweigh the disadvantages that may occur due to a less smooth search space topology. Both large and small neighbourhood sizes should generally be avoided so determining the best size neighbourhood is often a case of experimentation on search space, neighbourhood scheme and the problem being solved. Interestingly Jain *et al.* [6] demonstrated that for some problems a problem specific neighbourhood can developed which allow a very small neighbourhood size without decreasing search performance.

The level of independence between neighbours in the search space needs to be considered when assessing the effect of reducing the number of neighbours considered at each iteration. Neighbours are said to be independent if the result of making change A then change B is the same as making change B then change A. When neighbours are independent reducing the neighbourhood size may have little effect on the progress of the search as the independence of the moves simply means the search fine tunes different parts of the solution at each stage and less care is required when selecting moves. However if the moves are interdependent then the search performance may be adversely affected by reducing the neighbourhood size as the search could easily deviate from the path that would lead to the optima very easily. Again using the TSP problem and a 2-opt neighbourhood scheme is an example of a search space that is very independent. The changes made are localised to a small part of the solution, and therefore changes made to one region of the solution are not highly related to changes in another region of the solution. In this type of problem context a solution could be improved by taking its parts and improving them separately. In interdependent formulations the entire solution needs to be consider in its entirety.

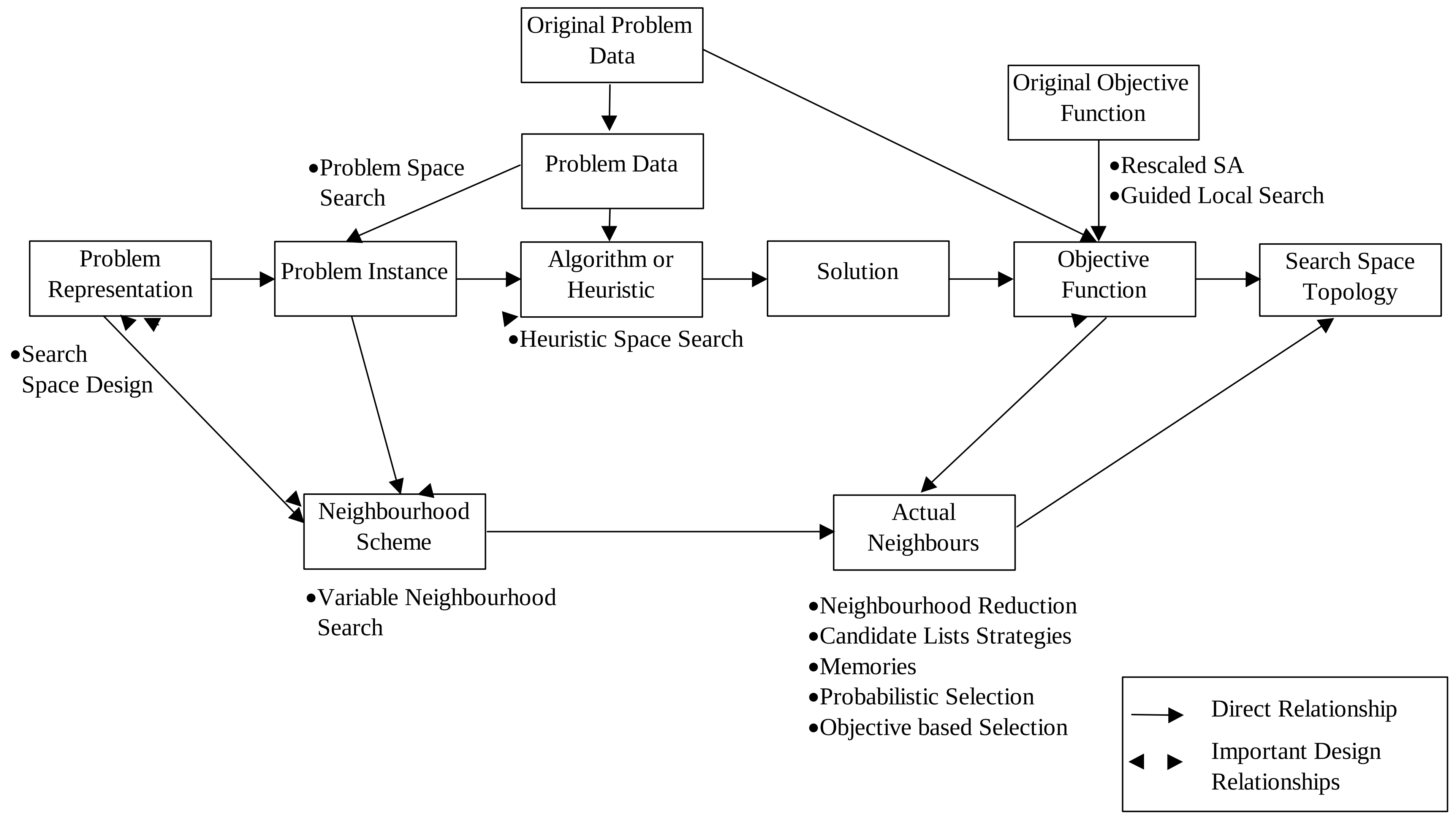
Original Problem Data
Problem Data
Original Objective Function
•Rescaled SA
•Guided Local Search
•Problem Space Search
Problem Representation
Problem Instance
Algorithm or Heuristic
Solution
Objective Function
Search Space Topology
•Heuristic Space Search
•Search Space Design
Neighbourhood Scheme
Actual Neighbours
•Variable Neighbourhood Search
•Neighbourhood Reduction
•Candidate Lists Strategies
•Memories
•Probabilistic Selection
•Objective based Selection
Direct Relationship
Important Design Relationships

Figure 1

The two direct influences on search space topology are the objective function – by calculating the quality of a particular search space instance on the search space – and the actual neighbours to that search space instance which effectively define what the relationships are between search space instances.

All of the relationships described above are summarised in Figure 1. Most of the elements outlined in the Figure 1 have been used in some way in order to either implement a search heuristic or to improve the performance of a search heuristic. For example:

- Storer *et al.* [16] looks at modifying both the problem data and the algorithm or heuristic as a means of moving the search from one solution to another. Problem instances in this case become either the problem data which is modified or the weightings given to particular heuristics.
- James [7] and James and Buchanan [8] look at changing the search space directly by representing the problem using a more abstract search space and a heuristic which converts this to an actual solution.
- The Neighbourhood scheme is one of the crucial in terms of defining the solution space as it defines the relationships between points. The emphasis here is on using an appropriate neighbourhood scheme for the problem that is being developed. The neighbourhood scheme defines the local minima in the solution space. Hence techniques such as "Variable Neighbourhood Search" [12] have been used in order to allow a search to break out of a local minima by switching from one neighbourhood scheme to another once a local minima is found.
- Methods that control the actual neighbours are probably the most widely used techniques in order to differentiate the various search heuristics and their enhancements. One aspect of this are techniques which convert a large neighbourhood scheme to smaller more manageable size, hence enabling the search to decide on which neighbour to move to more quickly. The main search techniques, such as Tabu Search and Simulated Annealing, could also be classified as primarily using this element in order to break out of local optima. These methods restrict the possible candidate moves that can be considered at a particular point during the search. In the case of Tabu Search the tabu list forms a memory which restrict the choice of neighbour while in the case of simulated annealing non-improving moves are only available probabilistically.
- Objective functions have also been used as a means for generating different solutions. Guided Local Search [19] is a technique that once the local search has reached a local minima, a penalty is added to the objective function of the search which deters the search from visiting that solution again. This effectively lifts that part of the solution space in such a way that is no longer is a local minima and hence allows the local search to progress again. Another heuristic which has used the manipulation of the objective function to aide the search is Rescaled Simulated Annealing proposed by Herault [4]. This search heuristic does not use objective function changes to make new neighbourhoods, as Guided Local Search does, but uses it to smooth out the solution space by redefining the energy or objective function by including a target energy level. The new energy function is defined as:

$$E_i = \left(\sqrt{f_i} - \sqrt{f_{target}}\right)^2$$

where E_i is the new objective function for solution i, f_i is the original objective function for solution i and f_{target} is a target energy level that dictates the level of smoothing and is directly related to the temperature of the SA.

The key in designing a search heuristic is to be aware of the interconnectivity between each of the different elements. Focusing solely on one element may not reveal the underlying changes happening in the overall structure of the search as changing one element can have a major impact on the performance of other elements. Understanding the interrelationships between different elements of the search enables us to start understanding why a particular change in a search heuristic causes a positive or negative change in the overall performance of the search.

4. Conclusions

The purpose of the framework is to give some insight into how the performance of search heuristics can be impacted by search space topology and to understand the impact of relationships between differing elements on search space topology. For example, too often the literature has developed search heuristics using a particular search space, neighbourhood scheme, etc. and found that this performed better on the standard benchmark problems than another standard heuristic or other search technique implemented. The question that needs to be answered is why does it perform better? Which improved relationship described in the model is causing this performance improvement? Does this hold for all types of problem data as opposed to just the standard benchmark problems? By identifying the elements and the interactions between them we can start to understand why differing elements may contribute to better performance of the search heuristic overall. This framework which concentrates on search space topology provides a means of structuring the relationships between elements.

The framework, by classifying the current literature according to the framework, also identifies some unexplored areas which may provide some new insights in the area of search heuristics. The first is the use of problem modification feeding into the objective function. In a similar manner to Guided Local Search which manipulates the objective function to bring the search out of a local minima, it may be possible to manipulate the problem data itself to get a similar effect. Whether this is a more effective way of dealing with local minima than Guided Local Search or any other technique is an open question.

A second opportunity is similar in that it is manipulating the problem data but unlike the Problem Space search the data would not be used as a search space but would be used by the algorithm or heuristic in order to manipulate the search space. It may be necessary in this instance to also calculate an actual solution using the original problem data in order to record best solutions to the original problem as well, however the search space surface would be governed by the modified problem data.

References

[1] J.W. Barnes, M. Laguna, *Solving the multiple machine weighted flow time problem using tabu search*, IIE Transactions, 25 (1993), pp 121-128

[2] J.R. Evans, *Structural Analysis of Local Search Heuristics in Combinatorial Optimization*, Computers and Operations Research, 14 (1987), pp 465-477

[3] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997

[4] L. Herault, *Rescaled Simulated Annealing – Accelerating Convergence of Simulated Annealing by Rescaling the States Energies*, Journal of Heuristics, 6 (2000), pp 215-252

[5] J.N. Hooker, *Testing Heuristics: We have it all wrong*, Journal of Heuristics, 1 (1995), pp 32-34

[6] A.S. Jain, B. Rangaswamy, S. Meeran, “New and “Stronger” Job-Shop Neighbourhoods: A Focus on the Method of Nowicki and Smutnicki (1996)”, *Journal of Heuristics*, 6 (2000), pp 457-480

[7] R.J.W. James, *Using Tabu Search to Solve the Common Due Date, Early/Tardy Machine Scheduling Problem,* Computers and Operations Research, 24 (1997), pp 199-208

[8] R.J.W.James, J.T. Buchanan, *A neighbourhood scheme with a compressed solution space for the early/tardy scheduling problem*, European Journal of Operational Research, 102 (1997), pp 513-527

[9] R.J.W.James, J.T. Buchanan, *Performance enhancements to tabu search for the early/tardy scheduling problem,* European Journal of Operational Research, 106 (1998), pp 254-265

[10] E. K. Karasakal, M. Köksalan, *A Simulated Annealing Approach to Bicriteria Scheduling Problems on a Single Machine*, Journal of Heuristics, 6 (2000), pp 311-327

[11] S. Kirkpatrick, C.D. Gelatt Jr, M.P. Vecchi, *Optimisation by Simulated Annealing,* Science, 220 (1983), pp 671-680

[12] N. Mladenovic, P. Hansen, *Variable neighborhood search*, Computers and Operations Research, 24 (1997), pp 1097-1100

[13] G.L. Nemhauser, L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1988

[14] C.R. Reeves, *Improving the efficiency of Tabu Search for machine sequencing problems*, Journal of the Operational Research Society, 44 (1993), pp 375-382

[15] L. Siklóssy, E. Tulp, *The Space Reduction Method: A method to reduce the size of search spaces*, Information Processing Letters, 38 (1991), pp 187-192

[16] R.H. Storer, S.D. Wu and R. Vaccari, *New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling*, Management Science, 38 (1992), pp 1495-1509

[17] S. Tsubakitani, J.R. Evans, *A two Dimensional Mapping for the Travelling Salesman Problem*, Computers and Mathematical Applications, 26 (1993), 65-73

[18] R.J.M. Vaessens, E.H.L. Aarts and J.K. Lenstra, *A Local search Template*, Computers and Operations Research, 11 (1998), pp 969-979

[19] C. Voudouris, E. Tsang, *Guided local search and its application to the travelling salesman problem*, European Journal of Operational Research, 113 (1999), pp 469-499