Tabu Search Scheduling

*Note: Sub-titles are not captured in Xplore and should not be used

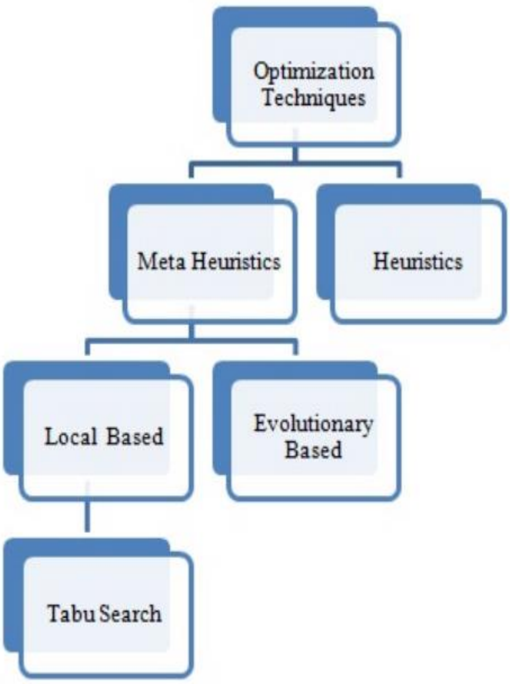
Muhammad Iqbal Bin Mohd Fauzi
Hochschule Hamm Lippstadt
Muhammad-iqbal.bin-mohd-fauzi@stud.hshl.de
Bruderstrasse 46, 59494 Soest

Abstract— **This paper presents one of the successful algorithms for randomized search, known as the tabu search scheduling algorithm, which is an example of an optimization technique. The Tabu search algorithm has its own characteristics that will aid in the solution of combinatorial optimization problems. To improve efficiency, pseudocode for the algorithm and methods for finding the best possible path are being investigated. For example, to reduce time and achieve global optimum. This paper also discusses an application that has successfully used this algorithm.**

Keywords—component, formatting, style, styling, insert (key words**)**

I. INTRODUCTION (*HEADING 1*)

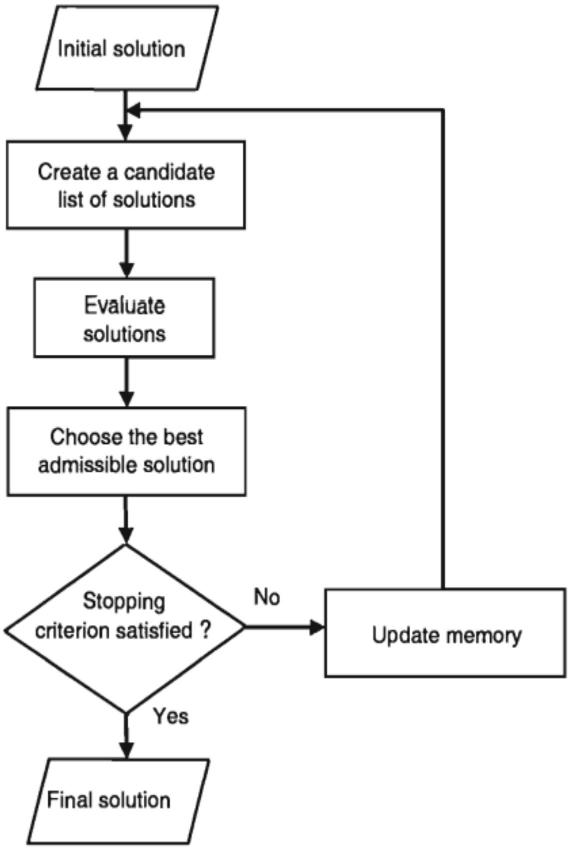
Recently, most of the engineers are having a lot of problems with combinatorial optimization. There are two types of optimization techniques in solving and tackle this problem. This optimization technique is classified in two part which are called metaheuristic and heuristic. Tabu search scheduling is one of the examples of metaheuristic algorithm and one of the successful techniques in finding optimal solution to solve combinatorial optimization problems using local search method [meta baru]. The examples of combinatorial optimization are vehicle routing problem, open vehicle routing problem, job shop problem, travelling salesman problem etc. [iee]. Tabu search was origin introduced by Fred Glover in 1986 through an article [meta]. Three main strategies are being used in a simple tabu search algorithm which are called forbidding strategy, freeing strategy and short-term energy [intelligent]. All these three strategies have their own functions in achieving the optimal solutions for the problems. For example, in forbidding strategies, candidate list or can be also called as tabu list will be influenced by the forbidding strategies which it will determine what will enter the list [intelligent]. After all the possible solution has enter the tabu list, freeing strategy will play role in controlling what will exit the tabu list and the time the possible solution will exit the list. The best solution strategy in choosing the admissible solution is it has better improvement from the current solution and satisfying the aspiration criteria. The main objective of this algorithm is to avoid repeatedly visiting the local optimum before finding the global optimum. To achieve that, ts will be using some memory to store the data of possible solution and the visited neighborhood until it finds the best solution for the problem



Classification of Optimization Techniques

II. ASPIRATION CRITERIA AND TABU RESTRICTION

The aspiration criteria are important in determining whether the tabu search procedure should be terminated. For example, if the solution is completely adequate for the problem, the procedure will be terminated. The term "sufficient solution" refers to when a solution can avoid the cycling process. The idea behind a tabu restriction is to allow the procedure to discover a specific search space. It means that if the solution adheres to the rules of the tabu restriction, it will be considered a move. Nonetheless, if the solution met the aspiration criterion, it will be considered admissible regardless of its tabu status. Aspiration criteria, which can be time-dependent or time-independent, also play an important role in providing a guideline for the search process. In time independent, if the solution demonstrates superior performance, resulting in a better solution, the tabu



classification will be removed from the trial solution. In short, aspiration moves are required to avoid prohibit moves while also allowing tabu moves that provide a better solution than the current solution. If a solution meets the aspiration criteria, it is admissible. For example, when a tabu search procedure has completed several iterations and the best solution has been found, it will be terminated. This is also can be called as stopping criterion.

Generally, there are few conditions of stopping criterion:

- Iteration number has reached the maximum.
- In each number of iterations, the best solution cannot be improved.
- The optimal solution was found. [iee improved tabu]

1. **begin TS**
2. TS_list = [];
3. S = initial solution;
4. S* = S;
5. **Repeat Step 2 to step 4**
6. Find the best admissible solution S1 belongs to Neighborhood of S
7. **if** f(S1) > f(S*) **then** S*:= S1;
8. S:= S1;
9. Update TS list TS_list;
10. **Until** the process, stopping criterion;
11. **End;**

III. TABU SEARCH ALGORITHM AND TABU LIST

The basic idea behind using the tabu search algorithm is that it will begin with the initial solution and progress through its neighborhood until it finds the optimal solution, which is the order's shortest distance. To achieve the best results from the tabu search algorithm, one aspect and concept must be clearly understood. This is known as a tabu list, and it can also be referred to as a candidate list. The tabu list represents all possible solutions. The tabu list is critical for avoiding cycling problems by preventing movement that may occur. The tabu list will keep track of all the moves that have been made. The purpose of doing this is to prevent future moves from being added to the list if the neighborhood is already being analyzed or visited [new metaheuristic]. After the candidate solution has been created, it will be evaluated. If a new or better solution is discovered, the old one will be deleted and replaced with the new one, making the new solution admissible. After the evaluation process, the procedure is not yet complete. If the stopping criterion is met, the search will be ended, and this will be the final solution.

Even if the solution is the best possible, if it does not meet the stopping criterion, it will not become the final solution, and the procedure will be restarted. The flow of the tabu search algorithm is show in fig..

IV. NEIGHBORHOOD AND MOVES

Modification of the solution will occur in each iteration of the solution. For example, some changes will be made from iteration 1 to iteration 2 of tabu search. A move is a change made to a solution. In general, the move can be accomplished by removing a specific neighborhood from a position in the sequence and replacing it with a new neighborhood in the same position and sequence. Neighborhood can be permuted in three ways: inversion, transposition, and displacement. The inversion works by swapping the locations of two elements that are adjacent to each other. Transposition occurs when two elements in a sequence swap location, and displacement occurs when one of the elements moves to another location in the sequence [meta lama]

The example on how the neighborhood can be permuted is shown on figure…

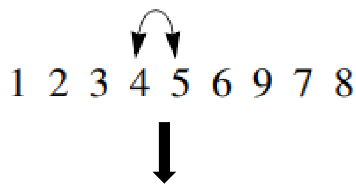


Fig 1: Ksdfk

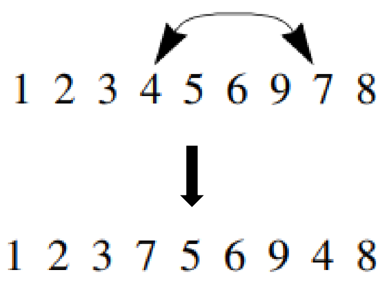


Fig 2: Dfjdk

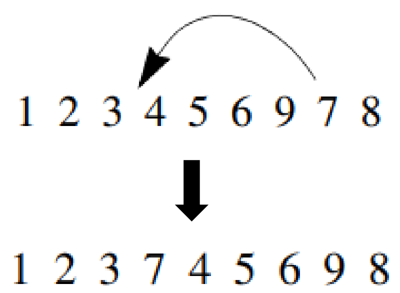


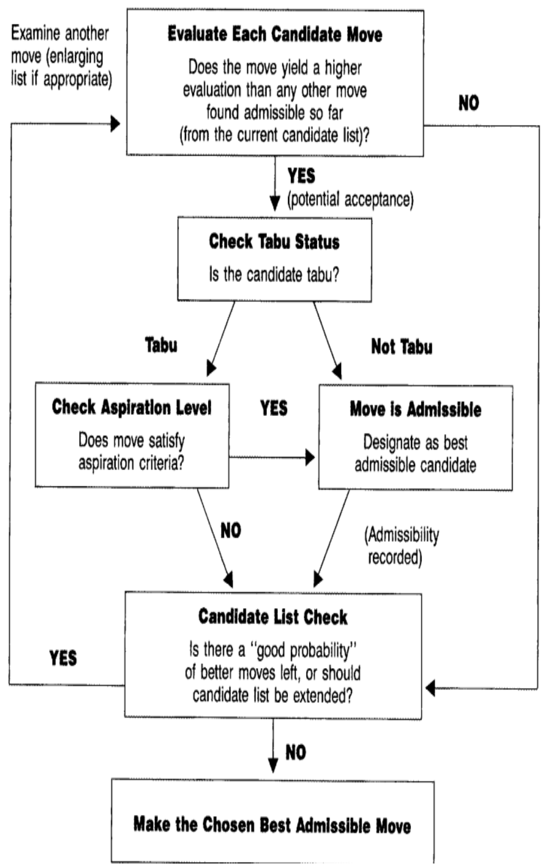
Fig 3: Ggg

V. DIVERSIFICATION

The tabu search algorithm places a high value on diversification. Because, as previously stated, aspiration criteria aid in the discovery of the best possible solution, but diversification aids in preventing the new neighborhood from becoming stuck at the local optima. The process is that it will collaborate with new neighborhoods that were previously not included in the solution, forcing the algorithm to search for areas that have not yet been explored. Diversification is essentially a strategy for exploring a given search space and expanding the search in new directions. This process will also make use of long-term memory, which will store all iterations from the beginning of the search to the current solution. As a result, the best solution is discovered.

VI. EVALUATIONN OF THE CANDIDATE LIST

The procedure for determining the best candidate move is straightforward. The first step is to determine whether the weight of the next candidate move will be higher or lower than the weight of the current neighborhood list. If it is higher, the next step would be to check the move's tabu status. It will have two options here. If it is not tabu, then the move is permissible. On the other hand, if it is tabu, the next step would be to check the aspiration criterion. Following both steps, the candidate list will be reviewed. In that order, the best candidate on the list is the one with no remaining options for a better move, and it is chosen as the best admissible move. Below shows the flow on how the candidate is being selected.



.

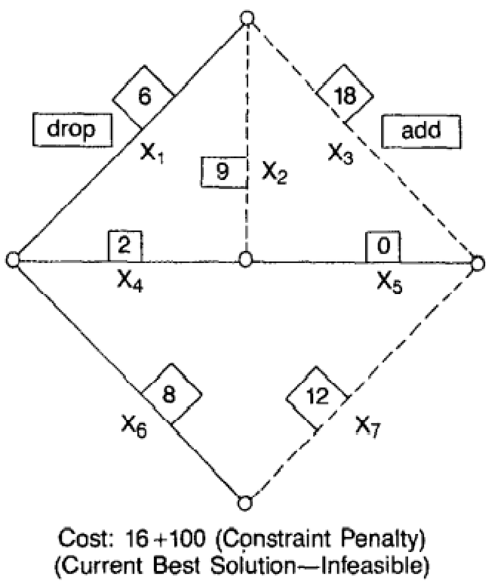
VII. ILLUSTRATION OF TABU SEARCH

A cost spanning tree problem can be used to demonstrate a simple example of tabu search. In this problem, the limitation will be to find a solution that can be described using edges. This problem has two limitations. For starters, the restriction will prevent certain edges from entering the tree. Second, one edge only exists if the other edge exists. If this limitation is violated during the iteration process, a penalty will be imposed. The solid line represents the tree's edges, while the dotted line represents the other edges. The cost of the edges is shown in a box above the edge, and the name of the edges is shown below the edges. The edges are labeled X1, X2, X3, X4, X5, X6, and X7. As per mentioned before, the limitation for this example is as below shown in figure…

No	Limitation
1	Edge of x1, x2 and x3 can be in three at most one.
2	X1 is only permitted in the tree if edge x3 also exist in the tree.

If the limitation is violated, a penalty of cost 50 will be imposed. Swap edges will be used to apply the tabu search. This swap edges process will involve adding and dropping edges to create a new solution. The rules of adding and dropping must be followed, and if it wants to drop an edge, it must be in a cycle where the edge is added. For example, if x2 wants to be added to the tree in iteration 2, the dropping will be on x3 or x5. If id x6 is added in iteration 3, the dropping will occur on x7, x5, or x4. In the case of a tabu restriction, an edge will become a tabu after being added to the tree, and no more than two tabus can exist at the same time. Furthermore,

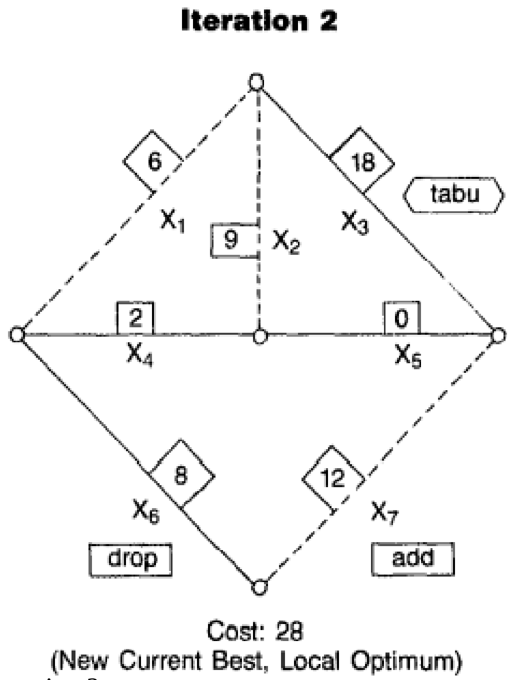
tabu status can only be retained for two iterations. The iteration that has been mentioned is shown below.



A. Iteration 1

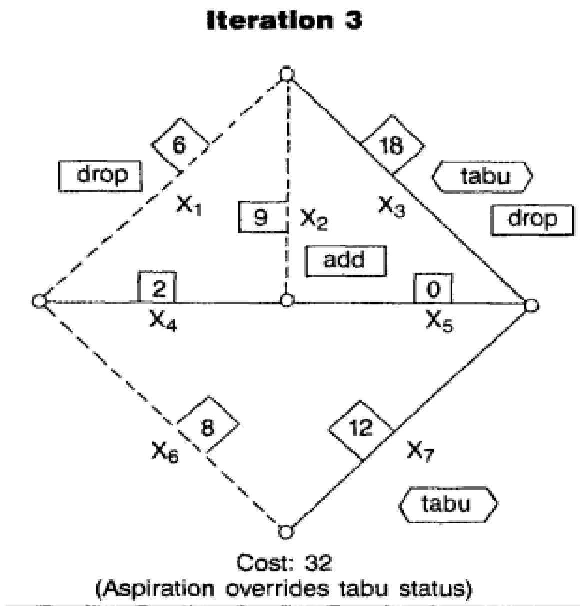
In this iteration, the best possible solution using the adding and dropping edges method is to add x3 and drop x3. However, both of the previously mentioned limitations are being violated. As can be seen, there are x1 and x6 in the tree, which are not permitted. Furthermore, edge x1 exists in the tree while edge x3 does not. Following that, the penalty will be imposed for these violations, totaling 116 (100+16). The penalty will be reduced from 100 to 10 in the next move, as no limitation is broken, as indicated in iteration 2.

dfgdg



B. Iteration 2

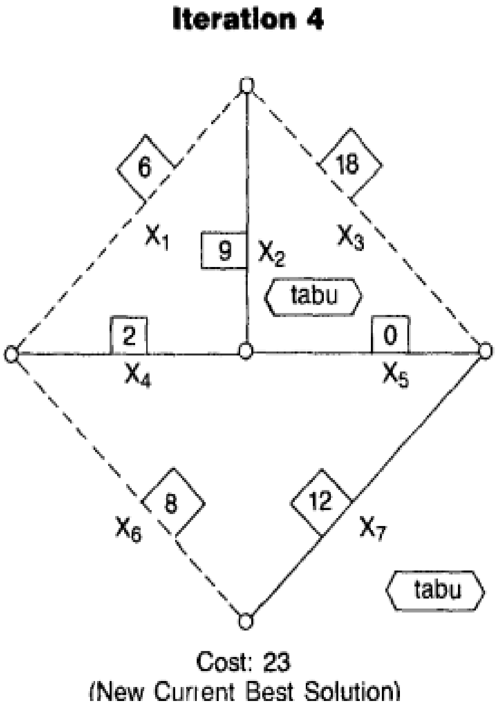
Because x3 is being added in this iteration, it will be necessary to add tabu status to this edge. The lowering of the edge is also included. The x2 will be dropped edges in this case. Because, as previously stated, x1 is in the x3 cycle. No penalty will be imposed in this iteration because both limitations are met. It will then lead to the current best solution and achieve the local optimum. This iteration's total cost will be 28, which is significantly less than iteration 1. The next possible move is to add edge x7 and drop x6.



dd

C. Iteration 3

This iteration's added edge is x7, and the dropping edge is x6. Because edge x7 has been added, the edge has now become tabu. However, this move will result in a violation of the restriction that no x1, x2, or x6 are in the edges. Despite the fact that a violation occurred, this move still meets the aspiration criterion. This demonstrates that, even if the move is illegal, no penalty will be imposed because it meets the aspiration criteria. This move will cost 32, which is higher than the cost in iteration 2. This move is still permissible because there is no possible solution that will lead to a better solution from iteration 2 because it is a local optimum.



df

D. Iteration 4

For this iteration, x2 edge will be added to the tree, resulting in a tabu, and x3 edge will be dropped. The edge x3 is tabu in the previous iteration, which is iteration 3. Due to the rule that a tabu can only exist on the edge for two iterations, the edge x3 is no longer tabu in this iteration. We now have the best solution available.

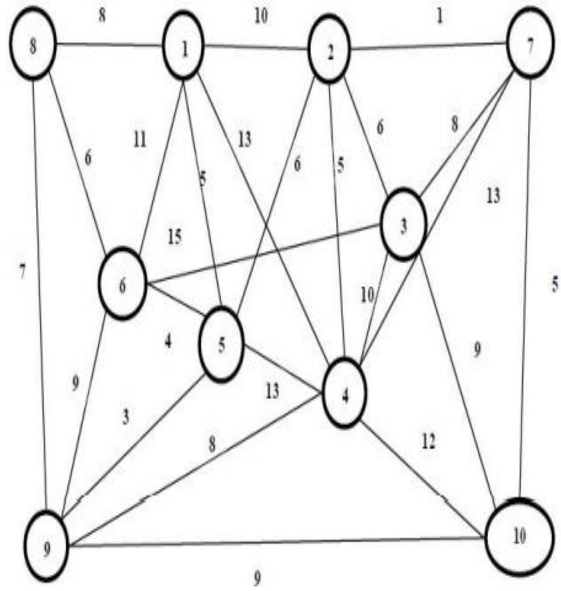
VIII. Application of Tabu Search

There are numerous applications that make use of the tabu search algorithm. The travelling salesman problem is one of the most popular applications that use the tabu search algorithm.

A. Travelling salesman problem

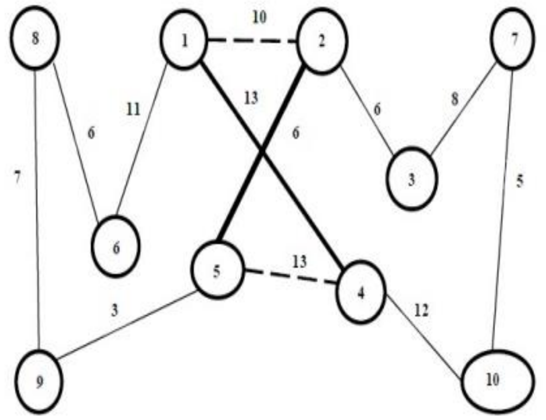
Ten nodes will be used in this example. The basic idea behind the travelling salesman problem is that each of the ten nodes in the tree must be visited at least once while maintaining the optimal solution. A tree can be symmetric or asymmetric. The term symmetric refers to the fact that the cost of moving from one node to another is the same both ways. For example, the cost from node 2 to node 1 is the same as the cost from node 1 to node 2. The term asymmetric refers to the inverse, in which the cost from node 1 to node 2 is not the same as the cost from node 1 to node 2.

The initial solution to the traveling salesman problem is shown in Figure 4.1. The edge's weight is indicated on the edge line.



For iteration 1, the move is made by swapping the edges that include the nodes 1,2,4, and 5. Instead of the edge from node 1 to node 2, it will now be replaced with the edge from node 1 to node 4. The same transfer is performed from node 5 to node 2. This makes the tour length 77.

For the second iteration, the same move will be used where the edge from node 3 to node 7 now be replace to the edge from node 3 to node 10. Node 7 will have an edge with node 4. This will yield the tour length to become 79 which is higher than iteration 1. This process is repeated until the best solution is discovered.



Tour Length 79

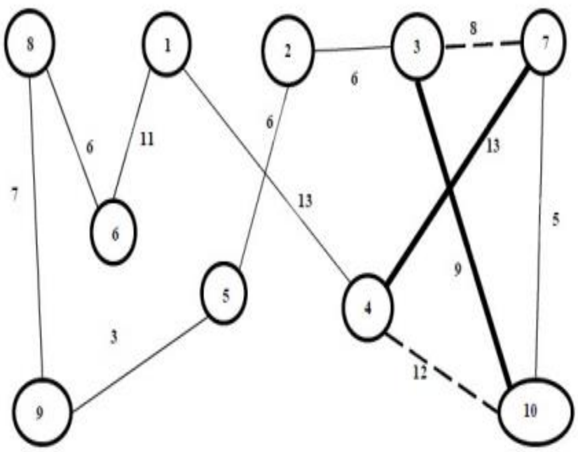


Fig. 4.4. Second iteration of tabu search

Tour Length 78

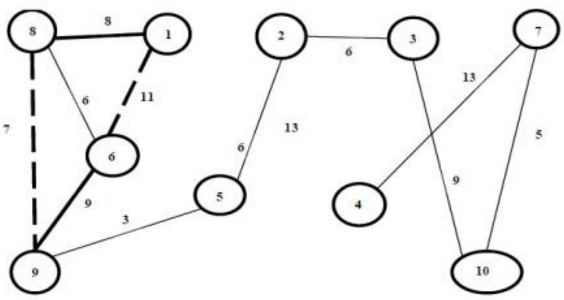


Fig. 4.5. Third iteration of tabu search

REFERENCES

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] .Edelkamp, S. and Schrödl, S., 2012. *Heuristic search*. Amsterdam: Morgan Kaufmann.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.