Neighborhood Hybrid Structure for Discrete Optimization Problems

Marco Antonio Cruz-Chávez, Alina Martínez-Oropeza, Sergio A. Serna Barquera
CIICAp, Universidad Autónoma del Estado de Morelos
Av. Universidad 1001. Col. Chamilpa, C.P. 62209. Cuernavaca, Morelos, México
{mcruz, alinam, aserna}@uaem.mx

Abstract

In this paper a comparative analysis of a neighborhood structures group are presented, including a hybrid structure, which arises of a combination of this set of structures. The efficiency and effectiveness of each structure was tested using the Classical Symmetric Travelling Salesman Problem. This study identifies the neighborhood structure that allows performing a better exploration and exploitation of the space solutions to discrete optimization problems. A neighborhood hybrid structure proposed has better performance comparing with other techniques, this is experimentally proved; in addition a competitive efficiency is shown.

1. Introduction

Many problems treated within the Combinatorial Optimization area require a search of approximate solutions due to the complexity and nature of these. This kind of problems is known as NP-Complete [1]. Due to the hardness of such problems, heuristic techniques have been used to neighborhood search structures, which have shown to be efficient methods searching approximated solutions for these problems.

In this paper five different neighborhood search functions are presented for the Classical Symmetric Travelling Salesman Problem. The essential part of a neighborhood is its size and structure [2]. While the neighborhood size is larger, the locally optimum solutions quality will be better, as well as the precision of final solution found, therefore, a heuristic with a large neighborhood will be more effective, otherwise, the time required to perform an iteration within the neighborhood will be increasing according to the size of neighborhood. According to this, finding an adequate neighborhood size it is required, which permits generate the best possible performance of the neighborhood function.

The choice of a neighborhood structure is performed by an analysis of effectiveness and efficiency of several structures to determine the type of structure to handle before it is implemented within meta heuristics such as Simulated Annealing, Iterated Local Search, Tabu Search [3], Memetic Algorithms [4], Ant Colony [5], among others. It allows that these meta heuristics works more efficiently searching global optimums in NP-Hard problems.

The hypothesis proposed in this paper is, given the complexity of solution space for discrete problems, it is possible that a hybrid structure get local optimums in a less complicated way, because it handles all the search features of each neighborhood structure that make it, allowing better solutions in a neighborhood, resulting in a better exploitation of the solution space for NP-Complete problems to reach a global optimum.

This paper is divided in the following sections. Section one is the introduction. Section two gives a general introduction to neighborhood features and define the Classical Symmetric Travelling Salesman Problem, which is used to test the performance of neighborhood structures under study. Section three presents the neighborhood search structures used in this research. Section four details the experimental testing, as well as the performance of neighborhood search techniques used. Section five presents the conclusions of this work.

2. Search by Neighborhood

A neighborhood is defined as a set of near solutions in an initial solution given, that is, given a feasible point $s \in S$ in an instance of a problem, a neighborhood of *s* is defined as a set *N(s)* of feasible points near *s*. The set *N(s)* called the neighborhood *s*, indicates that each solution $s' \in N(s)$ can be reached directly from *f* in one step. In accordance with this, the neighborhood is defined by the function $N : S \rightarrow 2^S$. To improve a solution s, it is necessary moving step by step from an initial feasible solution towards a solution that provides the minimum value of the objective function C, which usually involves costs.

The essential point of procedure is to start from a feasible point *s* and the set of solutions in *N(s)*, it is chosen an *s'* to improve *C* by a stochastic procedure, this is *C(s') <= C(s)*, with this, it is a replacement of the solution s and is positioned in this new point *s'*. So, each solution found that improve the before one considers the new provisional value of the objective function until the solution is no longer improved. The choice of an appropriates neighborhood structure is a critical aspect of designing this class of algorithms, so, choose that structure allowing a better exploration and exploitation of the solution space

Any optimization problem has a set, either finite or infinite number of possible solutions, according to its classification within the Complexity Theory. These types of local search algorithms are iterative, so, they start from an initial feasible solution and they are exploring the solution space to find neighbor solutions that improve the current solution, using several search strategies. That is, having a solution *s* given, any solution *s'* can be found from the solution *s*, through a performance called *movement*, which is the technique used to realize the exploration of the solution space. The kind of movement defines the structure and size of a neighborhood.

It is said that *s* is a local optimum if it is not a solution within *N(s)* that improves. The basic idea of the neighborhood structure is that, from an initial solution *s*, its neighbors *N(s)* are explored and evaluated to find a new better solution $s' \in N(s)$.

To determine the structure of a neighborhood, a neighborhood must be defined and the selection criteria of a neighbor. That is, if the selection criterion is true, a movement is realized, the process is repeated until the solution found can not be better, so, it is said to have reached a local optimum. Figure 1 shows the general algorithm for neighborhood search. The kind of movement to select a neighbor defines the neighborhood structure.

```
Generates initial solution s
Do
   s' = solution with σ movement.
   If  (f(s')<f(s)) then
      s' = the best solution so far
      s ← s'
    End-if
 While  Solution continues improving
```

Figure 1. General Algorithm for Neighborhood Search

The Combinatorial Optimization area and the Complexity Theory are an essential part of the environment of the Classical Symmetric Travelling Salesman Problem (CSTSP). The Discrete Optimization studies modeling and algorithmic solution of complex problems which seeks to maximize or minimize an objective function of several variables defined on a discrete set [1, 6]. Its application is distributed in different areas such as industry, logistics, computer science, engineering and business administration, this problem has been widely studied by researches around the world.

The Complexity Theory studies the resources required to solve a problem such as the *time* (number of steps for implementing an algorithm to solve a problem) and the *space* (amount of memory used to solve a problem) [7]. Based on the study of the above resources, the complexity theory classifies problems based on the difficulty to solve them.

The classification of complexity theory is divided into P problems (polynomial), NP (Nondeterministic polynomial) and NP-Complete.

The P class includes the problems which can be solved by a deterministic machine in polynomial time. NP problems are those can be bounded by a nondeterministic machine in polynomial time. There are plenty of problems classified as NP complete, one of the most studied by researchers is the Travelling salesman problem.

The Classical Symmetric Travelling Salesman Problem (CSTSP) consists in minimizing the distance traveled by Travelling salesman to visit every city once and returning to the home town [8]. The CSTSP is a well known problem within the combinatorial optimization area due to its classification as NP-Complete problem in the Complexity theory. For this reason we resorted to the use of approximate methods for finding near-optimal solutions in a reasonable computational time [6].

We can define a tour as a complete path that visits each city only once, so, each solution corresponds to a path. The solution space of the problem is given by a set of π cyclic permutations of the *n* cities.

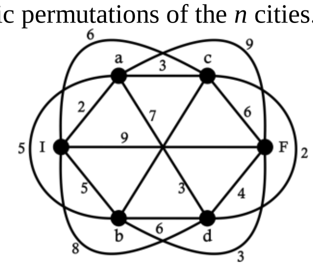

Figure 2. Undirected graph to the Classical Symmetric Travelling Salesman Problem

This problem can be modeled easily by an undirected graph which forms a clique, where each vertex represents each city to visit and edges are tours or distances in kilometers that the Travelling salesman has to travel going from city *i* to city *j* (Figure 2).

Let a graph $G = (V, E)$, where V defines the nodes or cities to visit, $V = \{1, \ldots, n\}$ and E represents edges or distances to travel from city i to city j, therefore $E = \{(i, j) : i, j \in V\}$, and let c_{ij} the associated cost with the edge (i, j), according to this, the Classical Symmetric Travelling Salesman Problem can be formulated as follows (Figure 3).

$$Min \sum_{(i,j)\in E} C_{ij} X_{ij} \qquad (1)$$

$$\sum_{\{C_j:(i,j)\in E\}} X_{ij} = 1 \qquad \forall i \qquad (2)$$

$$\sum_{\{C_i:(i,j)\in E\}} X_{ij} = 1 \qquad \forall i \qquad (3)$$

$$\sum_{\{(i,j)\in E, i\in S, j\in S\}} X_{ij} \le |S| - 1 \quad to \quad S \subset V, 2 \le |S| \le |V| - 2 \quad \forall i \qquad (4)$$

Figure 3. Mathematical Model of Integer Linear Programming for the Classical Symmetric Travelling Salesman Problem, Fischetti et al., in [9]

According to the mathematical model presented in Figure 3, constraints represents by (2), (3) and (4) specify that the starting point *i* is the same as the end of the tour, in addition, each city must be visited only once, in order to satisfy the objective function which is represented by (1) that minimizes the total cost of the tour.

3. Neighborhood Structures

A neighborhood search technique is implemented in order to improve a solution, which is necessary to move step by step from an initial feasible solution that provides the minimum value of the objective function *C* [10].

The difference between neighborhood search techniques lies in the type of movement that is performed iteratively to go from an initial solution to a better neighbor solution, until it can not be improved, then it is said to have reached a local optimum. The following explains the five neighborhood techniques used in this study.

3.1. Neighborhood Structure using an Adjacent Pair

An initial feasible solution s is generated for neighborhood search structure using an adjacent pair [11, 12, 13], from this solution, a random number *num* is chosen, it corresponds to a vector position where the initial solution is stored.

When the random position is already chosen, *num + 1* is taken in vector (Figure 4) and the permutation is realized between this adjacent pair of numbers to get a new neighbor solution *s'*.

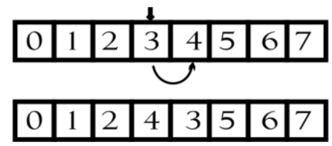

Figure 4. Adjacent permutation to a feasible solution from a position chosen randomly

3.2. Neighborhood Structure using a Random Pair

The same procedure used for an adjacent pair is applied to carry out a neighborhood search using a random pair [1, 12, 14], the difference is that two random numbers n and n_1 are generated, those numbers must not be adjacent in the solution vector (Figure 5).

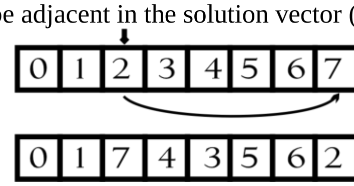

Figure 5. Random permutation in a feasible solution from two positions selected randomly

3.3. Neighborhood Structure using two Adjacent Pairs

In case of a permutation using two adjacent pairs [15], it was necessary to generate randomly two numbers that satisfy the following (Figure 6): they must be different numbers and each adjacent pair must not be adjacent with another pair.

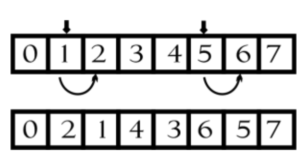

Figure 6. Two adjacent permutations from two positions selected randomly.

3.4. Neighborhood Structure using two Random Pairs

A neighborhood search technique that uses two random pairs [14, 16, 17], requires the generation of four random numbers, which must be different and nonadjacent (Figure 7).

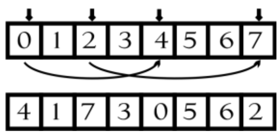

Figure 7. Two permutations of random pairs using four positions in randomly form

3.5. Neighborhood Hybrid Structure

The neighborhood hybrid structure proposed in this work is a mixture of the techniques explained previously; an adjacent pair, a random pair, two adjacent pairs and two random pairs. The type of permutation is chosen randomly during the algorithm execution (Figure 8).

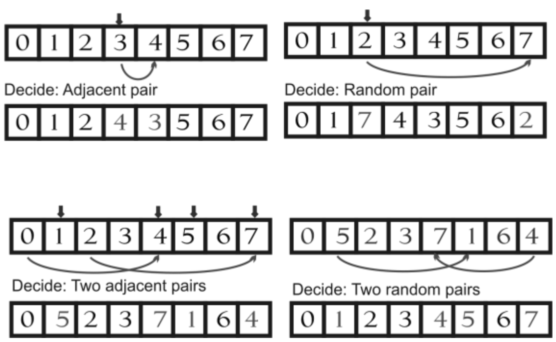

Figure 8. Before any exchange, the algorithm decides randomly what type of permutation will be realized (adjacent or random permutations).

4. Experimental Results

The experimental tests of neighborhood search techniques were realized in a laptop with Intel Centrino Core 2Duo processor at 2.0 GHz, 3 GB RAM and Windows Vista O. S. The compiler used was Visual C, 2008. The compiler used was Visual C, 2008. The test problems were randomly generated with 200, 500, 1000, 2000 and 4000 cities respectively.

Each neighborhood search technique conducted 30 executions of each instance size, using a stop criterion of 5 minutes for each test problem. Results obtained were saved in terms of the best solution, the number of times was found and the total number of iterations made during execution. The same test problems were executed in each neighborhood search technique, this allows performing a direct comparison with the results and conduct a more reliable analysis.

4.1 Effectiveness Tests

Tables from 1 to 5 present the results obtained with the instances of CSTSP executed in the neighborhood search techniques, to evaluate the average cost function and its standard deviation σ for 30 tests in each instance. It is observed that the neighborhood hybrid search technique is better in effectiveness in most of the instances of CSTSP, there is an exception in the instance of 500 cities where the hybrid technique is in second place of effectiveness.

In order of effectiveness, the second most effective technique uses permutations between two random pairs. The standard deviation behavior to hybrid technique shows that the dispersion of the solutions is higher than the other techniques, this means that solutions of the 30 tests are different depending of the instance where it is executed, therefore it has a greater exploitation of the solution space, because it has access to a wider range of solutions, due to the different type of movements (adjacent and nonadjacent), which are realized randomly.

Tables from 1 to 5 also present the results of the instance evaluated, as hypothesized, the best solution was obtained by the neighborhood hybrid structure technique, this indicates that a hybridization can get better solutions; it is shown either the best solution or the worst solution. The worst solution is one of the lowest comparing results of the other techniques. There is an exception in the instance of 1000 cities where the worst solution is one of the poorest found.
cities.

Table 1. Results for 200 cities

Structure	Best	Worst	Average	σ
Adjacent pairs	1913	2128	2012	58
Random pairs	1871	2093	1994	61
Two adjacent pairs	1865	2076	1974	48
Two random pairs	1852	2041	1945	51
Hybrid	1792	2048	1945	64

Table 2. Results for 500 cities

Structure	Best	Worst	Average	σ
Adjacent pairs	5006	5248	5161	58
Random pairs	4953	5228	5118	70
Two adjacent pairs	5007	5235	5123	57
Two random pairs	4946	5176	5057	69
Hybrid	4942	5218	5098	70

Table 3. Results for 1000 cities

Structure	Best	Worst	Average	σ
Adjacent pairs	48565	50346	49661	525
Random pairs	48173	50376	49375	587
Two adjacent pairs	47820	50105	49202	579
Two Random pairs	47441	50527	49166	789
Hybrid	47424	50527	49128	795

Table 4. Results for 2000 cities

Structure	Best	Worst	Average	σ
Adjacent pairs	100095	101793	101075	520
Random pairs	99148	101555	100617	630
Two adjacent pairs	99136	101743	100467	705
Two random pairs	97795	101043	99856	819
Hybrid	97712	101615	100232	846

Table 5. Results for 4000 cities

Structure	Best	Worst	Average	σ
Adjacent pairs	196935	198776	198292	423
Random pairs	196627	198819	198092	575
Two adjacent pairs	196032	198341	197373	540
Two random pairs	195579	198289	197524	617
Hybrid	194498	198309	197280	875

4.2. Efficiency Tests

It was performed the time calculation to obtain 1000 solutions in each CSTSP instance for each neighborhood search technique. Figure 9 presents the performance of each technique as will increasing the instance size.

Figure 9 shows that in the range of 200 to 2000 cities in efficiency performance, the hybrid structure proposed is similar to other structures. Naturally, the behavior of interest is in case of large instances. In case of the instance of 4000 cities, the hybrid structure proposed is within a competitive range competitive with other neighborhood structures, because, although this is not what has the best efficiency, it is not the worst performance.

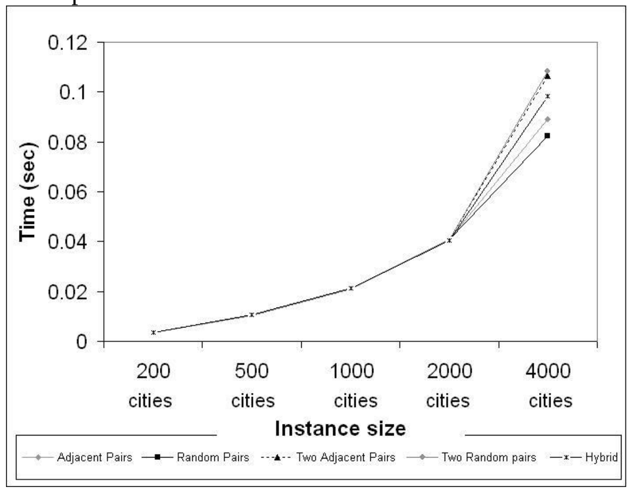

Figure 9. Results of each neighborhood structure taking 1000 solutions in a range of 200-4000 cities

Figure 9 shows, in case of 4000 cities, the efficiency of hybrid structure is in the middle comparing with other structures, that is, the efficiency order of the algorithms from the best to the worst is as follows, Hybrid, two random pairs, two adjacent pairs, random pair and adjacent pairs.

The efficiency of hybrid structure behavior is logical because it is composed of the other four structures and the implementation of each one in a hybrid is randomly, which makes that the hybrid structure efficiency is in the middle. Also, figure 1 shows that the time to find 1000 solutions increase considerably, due to the larger neighborhood necessarily need more time to be explored, thus, the response time to find 1000 solutions increases.

Future Work

Implementation of neighborhood hybrid structures for meta heuristic to evaluate the effectiveness and efficiency to improve the performance of these meta heuristics.

5. Conclusion

According to the results obtained in the efficacy analysis, the hypothesis of is proved that a neighborhood hybrid structure can more easily achieve better solutions. This is shown experimentally to perform an analysis of five types of neighborhood structures applied to the Classical Symmetric Travelling Salesman Problem which is considered a NP-complete problem. It is shown that the efficiency of neighborhood hybrid structure is practically the same behavior from the other four structures presented for instances of Classical Symmetric Travelling Salesman Problem in a range from 200 to 2000 cities. From 4000 cities, the hybrid structure efficiency decreases, being at the top of structures an instance of 4000 cities, the hybrid structure decreases its structures being at the top of structures. The permutation of two pairs in each iteration and below the structures that handle permutation of a single pair in each iteration.

Applying this kind of neighborhood hybrid structure to meta heuristic like Simulated Annealing, Memetic algorithms and another kind of heuristics that applies iterative local search, in NP-complete problems, suggest that would improve the effectiveness of these meta heuristics.

6. References

[1] Papadimitriou, C.H., Steiglitz, K. Combinatorial Optimization, Algorithms and Complexity. Dover Publications, Inc. Mineola, New York. USA ed. 1998.

[2] Ahuja Ravindra K., Ergun Özlem, Orlin James B., Punnen Abraham P. Study of Very Large Scale Neighborhood Searching Techniques. On-line: http://mit.ocw.universia.net/15.053/s02/ pdf/s02-lec23b.pdf. 2000.

[3] Michaelewicz Z. And Fogel D. B. How to Solve it: Modern Heuristics. Springer-Verlag Berlin Heidelberg New York. ISBN. 3-540-66061-5, pp. 117-125. 2000.

[4] Cruz-Chávez M. A, Díaz-Parra O., Juárez-Romero D., Martínez-Rangel M. G, Memetic Algorithm Based on a Constraint Satisfaction Technique for VRPTW, Lecture Notes in Artificial Intelligence, Springer Verlag Pub., Berlin Heidelberg, ISSN: 0302-9743, Vol. 5097, pp. 376-387, 2008.

[5] Alba E. Parallel Metaheuristics. A New Class of Algorithms. ISBN-13 978-0-471-67806-9 and ISBN-10 0-471-67806-6 (cloth). United States of America. pp. 172-174, 2005.

[6] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and approximation: Combinatorial Optimization Problems and their Aproximability Properties. Springer-Verlag. 1999.

[7] Cortéz-Augusto, Computacional Complexity Theory and Computability theory, Rev. Investig. Sist. Inform, INFOSYS, pp. 102-105. ISSN: 1815-0268, 2004.

[8] Liu S. B., Ng K. M., Ong. H. L., A New Heuristic Algorithm for the Classical Symmetric Traveling Salesman Problem, World Academy of Science, Engineering and Technology, 267-271, 27 2007.

[9] Fischetti M., Salazar-González J. J. y Toth P. The Symmetric Generalized Traveling Salesman Polytope. Journal NETWORKS, Vol. 26. Issue. 2 CCC 0028-3045/95/020113-11. pp. 113-123. 1995.

[10] Cruz-Chávez M. A, Rivera-López R., A Local Search Algorithm for a SAT Representation of Scheduling Problems, Lecture Notes in Computer Science, Springer-Verlag Pub., Berlin Heidelberg, ISSN: 0302-9743, Vol.4707, No. 3, pp. 697-709, 2007.

[11] Lin S. & Kernighan W. An Effective Heuristic for the traveling Salesman Problem. Bell Telephone Laboratories, Incorporated, Murray Hill, N.J. On line http://www.crema.unimi.it/~righini/Didattica/Algoritmi%20Euristici/MaterialeAE/Lin%20Kernighan%20TSP.pdf.,1971.

[12] González-Velázquez Rogelio, Bandala Garcés M. A. Hibrid Algorithm: Scaling Hill and Simulated Annealing to Solve the Cuadratic Allowance Problem. 3er Taller Latino Iberoamericano de Investigación de Operaciones. Acapulco, Guerrero, México, 2009.

[13] Pacheco J. y Delgado, C.: Diferent Experiences Results with Local Search applied to Path Problem. Revista Electrónica Rect@.ASEPUMA, vol.2, nº 1, pp. 54- 81. 2000.

[14] Kenneth D. B., Cost Versus Distance in the Traveling Salesman Problem. UCLA Computer Science Dept., Los Angeles. CA 90024 1596 USA, https://eprints.kfupm.edu.sa/ 32861/ 1/32861.pdf

[15] Lourenço H. R., Martin O. C., Iterated Local Search.http://arxiv.org/PS_cache/math/pdf/0102/0102188v1.pdf, 2001.

[16] Martin O., Otto S. W. and Felten E. W., " Large Step Markok Chains for the Traveling Salesman Problem" Complex Systems 5(3). pp 299-326, 1991.

[17] Martin O., Otto S. W. and Felten E. W., "Large Step Markov Chains for the TSP Incoporating Local Search Heuristics". Operations Res. Letters 11, pp 219-224, 1992.