

QUANTUM INSPIRED PARTICLE SWARM COMBINED WITH LIN-KERNIGHAN-HELSCGAUN METHOD TO THE TRAVELING SALESMAN PROBLEM

Bruno Avila Leal de Meirelles Herrera¹, Leandro dos Santos Coelho²
and Maria Teresinha Arns Steiner^{3*}

Received March 24, 2014 / Accepted May 2, 2015

ABSTRACT. The Traveling Salesman Problem (TSP) is one of the most well-known and studied problems of Operations Research field, more specifically, in the Combinatorial Optimization field. As the TSP is a NP (Non-Deterministic Polynomial time)-hard problem, there are several heuristic methods which have been proposed for the past decades in the attempt to solve it the best possible way. The aim of this work is to introduce and to evaluate the performance of some approaches for achieving optimal solution considering some symmetrical and asymmetrical TSP instances, which were taken from the *Traveling Salesman Problem Library* (TSPLIB). The analyzed approaches were divided into three methods: (i) Lin-Kernighan-Helsgaun (LKH) algorithm; (ii) LKH with initial tour based on uniform distribution; and (iii) an hybrid proposal combining Particle Swarm Optimization (PSO) with quantum inspired behavior and LKH for local search procedure. The tested algorithms presented promising results in terms of computational cost and solution quality.

Keywords: Combinatorial Optimization, Traveling Salesman Problem, Lin-Kernighan-Helsgaun algorithm, Particle Swarm Optimization with Quantum Inspiration.

1 INTRODUCTION

The Combinatorial Optimization problems are present in many situations of daily life. For this reason, and because of the difficulty in solving them, these problems have increasingly drawn the

*Corresponding author.

¹Pós Graduação em Engenharia de Produção de Sistemas (PPGEPS), Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, PR, e Fundação CERTI, Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil. E-mail: bruherrera@gmail.com

²Pós Graduação em Engenharia de Produção de Sistemas (PPGEPS), Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, PR, e Pós Graduação em Engenharia Elétrica (PPGEE), Universidade Federal do Paraná (UFPR), Curitiba, PR, Brasil. E-mail: leandro.coelho@pucpr.br

³Pós Graduação em Engenharia de Produção de Sistemas (PPGEPS), Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, PR, e Pós Graduação em Engenharia de Produção (PPGEP), Universidade Federal do Paraná, Curitiba, PR, Brasil. E-mail: maria.steiner@pucpr.br

attention of many researches of various fields of knowledge who have made efforts to develop ever efficient algorithms that can be applied to such problems (Lawler et al., 1985).

Some of the examples of Combinatorial Optimization problems are: The Traveling Salesman Problem (TSP), the Knapsack Problem, the Minimum Set Cover problem, the Minimum Spanning Tree, the Steiner Tree problem and the Vehicle Routing problem (Lawler et al., 1985).

The TSP is defined as a set of cities and an objective function which involves the costs of travelling from one city to the other. The aim of the TSP is to find a route (tour, path, itinerary) through which the salesman would pass all the cities one single time, giving a minimum total distance. In other words, the aim of the TSP is to achieve a route that will form a Hamiltonian cycle (or circuit) of minimum total cost (Pop et al., 2012).

The Combinatorial Optimization problems are used as models in many real situations (Robati et al., 2012; Lourenço et al., 2001; Johnson & McGeoch, 1997; Mosheiov, 1994; Sosa et al., 2007; Silva et al., 2009; Steiner et al., 2000), both in simple objective versions as in multiple criteria versions (Ishibuchi & Murata, 1998; Jaskiewicz, 2002). The Combinatorial optimization has led many researchers to pursue approximate solutions of good quality and even accept that they are considered insoluble in polynomial time.

On the other hand, many metaheuristic techniques have been developed and combined in order to solve such problems: Dong et al. (2012) combine both Genetic Algorithm (GA) and Ant Colony Optimization (ACO) together in a cooperative way to improve the performance of ACO for solving TSPs. The mutual information exchange between ACO and GA in the end of the current iteration ensures the selection of the best solutions for next iteration. In this way, this cooperative approach creates a better chance in reaching the global optimal solution because independent running of GA maintains a high level of diversity in next generation of solutions. Nagata & Soler (2012) present a new competitive GA to solve the asymmetric TSP. The algorithm has been checked on a set of 153 benchmark instances with known optimal solution and it outperforms the results obtained with previous ATSP (Asymmetric TSP) heuristic methods.

Sauer et al. (2008) used the following discrete Differential Evolution (DE) approaches for the TSP: i) DE approach without local search, ii) DE with local search based on Lin-Kernighan-Heulsgaun (LKH) method, iii) DE with local search based on Variable Neighborhood Search (VNS) and together with LKH method. Numerical study was carried out using the TSPLIB of TSP test problems. The obtained results show that LKH method is the best method to reach optimal results for TSPLIB benchmarks but, for largest problems, the DE+VNS improve the quality of the obtained results.

Chen & Chien (2011a) make experiments using the 25 data sets from the TSPLIB (*Traveling Salesman Problem Library*) and compare the experimental results of the proposed method (Genetic Simulated Annealing and Colony Systems with Particle Swarm Optimization, PSO) with many other methods. The experimental results show that both the mean solution and the percentage deviation of the mean solution to the best known solution of the proposed method are better than those methods. Chen & Chien (2011b) present in another paper another new

method (Parallelized Genetic Ant Colony System) for solving the TSP. It consists of the GA with a new crossover operation and a hybrid mutation operation, and the ant colony systems with communication strategies. They also make an experiment with three classical data sets from the TSPLIB to test the performance of the proposed method, showing that it works very well. Robati et al., 2012, present an extension of PSO algorithm which is in conformity with actual nature is introduced for solving combinatorial optimization problems. Development of this algorithm is essentially based on balanced fuzzy sets theory. The balanced fuzzy PSO algorithm is used for TSP.

Albayrak & Allhverdi (2011) developed a new mutation operator to increase GA performance in finding the shortest distance in the TSP. The method (Greedy Sub Tour Mutation) was tested with simple GA mutation operator in 14 different TSP instances selected from TSPLIB. The new method gives much more effective results regarding the best and mean error values. Liu & Zeng (2009) proposed an improved GA with reinforcement mutation to solve the TSP. The essence of this method lies in the use of heterogeneous pairing selection instead of random pairing selection in EAX (edge assembly crossover) and in the construction of reinforcement mutation operator by modifying the Q-learning algorithm and applying it to those individual generated from modified EAX. The experimental results on instances in TSPLIB have shown that this new method could almost get optimal tour every time, in reasonable time.

Puris et al. (2010) show how the quality of the solutions of ACO is improved by using a Two-Stage approach. The performance of this new approach is studied in the TSP and Quadratic Assignment Problem. The experimental results show that the obtained solutions are improved in both problems using this approach. Misevičius et al. (2005) show the use of Iterated Tabu search (ITS) technique combined with intensification and diversification for solving the TSP. This technique is combined to the 5-opt and, the errors are basically null in most of the TSPLIB tested problems. Wang et al. (2007) presented the use of PSO for solving the TSP with the use of quantum principle in order to better guide the search for answers. The authors make comparisons with Hill Climbing, Simulated Annealing and Tabu search, and show an example in which the results of 14 cities are better this technique than with others.

However, on the other hand, during the last decade, the quantum computational theory had attracted serious attention due to their remarkable superiority in computational and mechanical aspects, which was demonstrated in Shor's quantum factoring algorithm (Shor, 1994) and Grover's database search algorithm (Grover, 1996). One of the recent developments in PSO is the application of quantum laws of mechanics to observe the behavior of PSO. Such PSO's are called quantum PSO (QPSO).

In QPSO, the particles are considered to lie in a potential field. The position of each particle is depicted by using a wave function instead of position and velocity. Like PSO, QPSO is characterized by its simplicity and easy implementation. Besides, it has better search ability and fewer parameters to set when compared against PSO (Fang et al., 2010). A convergence analysis of QPSO is presented in Sun et al. (2012).

Comprehensively reviews of applications related to quantum algorithms and QPSO in the design and optimization of various problems in the engineering and computing fields have been presented in the recent literaturesuch as Fang et al. (2010) and Manju & Nigam (2014). Other recent and relevant optimization applications of the QPSO related to continuous optimization can be mentioned too, such as feature selection and spiking neural network optimization (Hamedl et al., 2012), image processing (Li et al., 2015), economic load dispatch (Hosseinezhad et al., 2014), multicast routing (Sun et al., 2011), heat exchangers design (Mariani et al., 2012), inverse problems (Zhang et al., 2015), fuzzy system optimization (Lin et al., 2015), neuro-fuzzy design (Bagheri et al., 2014) and parameter estimation (Jau et al., 2013).

In terms of combinatorial optimization, this paper introduces a new hybrid optimization approach of Lin-Kernighan-Helsgaun technique (LKH) and quantum inspired particle swarm algorithm for solving someTSP symmetric and asymmetric instances of the TSPLIB.

The rest of this paper is organized as follows: section 2 introduces a brief history of the TSP, as well as some ways of solving it. Section 3 details the techniques approached in this paper: Lin-Kernighan (LK); Lin-Kernighan-Helsgaun (LKH); PSO and QPSO. In Section 4, the achieved results are tabulated with the implement of the approached techniques for the instances of the TSLIB repository involving symmetric and asymmetric problems, and finally, section 5 presents the conclusions.

2 LITERATURE REVIEW

It is believed that one of the pioneer works about TSP was written by William Rowan Hamilton (Irish) and Thomas Penyngton Kirkman (British) in the XIX century based in a game in which the aim was to “travel” through 20 spots passing though only some specific connections allowed between the spots. Lawler et al. (1985) report that Hamilton was not the first one to introduce this problem, but his game helped him to promote it. The authors report yet that in modern days the first known mention to the TSP is attributed to Hassler Whitney, in 1934, in a work at Princeton University. Gradually, the amount of versions of the problem increased and some of them show peculiarities that make their resolution easier. In the TSP, a trivial strategy for achieving optimal solutions consists in evaluating all the feasible solutions and in choosing the one that minimizes the sum of weights. The “only” inconvenience in this strategy is in the combinatorial explosion. For example, for a TSP with n cities connected in pairs, the number of feasible solutions is $(n - 1)!/2$.

The TSP can be represented as a permutation problem. Assuming P_n a collection of all permutations of the set $A = \{1, 2, \dots, n\}$, the aim for solving the TSP is to determine $\Pi = (\Pi(1), \Pi(2), \dots, \Pi(n))$ in P_n so that

$$\left(\sum_{i=1}^{m-1} d(c_{\Pi(i)}, c_{\Pi(i+1)}) \right) + d(c_{\Pi(m)}, c_{\Pi(1)})$$

will be minimized. For example, if $n = 5$ and $\Pi = (3, 4, 1, 5, 2)$ then the corresponding route will be (3-4-1-5-2).

Mathematically the TSP is defined as a decision problem and not an optimization problem. Thus, the fundamental answer that must be given is whether there is or not a route with a minimum total distance. In such context, the decision problem can be modeled as follows. Given a finite set $C = \{c_1, c_2, \dots, c_m\}$ of cities in $d(c_i, c_j) \in \mathbb{Z}^+$ distance for each pair of cities $c_i, c_j \in C$, and the total initial cost $T \in \mathbb{Z}^+$, where \mathbb{Z}^+ is the set of positive integers such as inequation (1) is met.

$$\left(\sum_{i=1}^{m-1} d(c_{\Pi(i)}, c_{\Pi(i+1)}) \right) + d(c_{\Pi(m)}, c_{\Pi(1)}) \leq T \quad (1)$$

According to Lawler et al. (1985), many techniques have been developed over the years in order to construct exact algorithms that would solve the TSP, such as Dynamic Programming, Branch-and-Bound and Cutting-plane methods (Papadimitriou & Steiglitz, 1982; Nemhauser & Wolsey, 1988; Hoffmann, 2000). However, the computational trimmings to solving large instances of exact algorithms are unfeasible/prohibitive.

Therefore, we use heuristic methods that are able, in a general way, to provide satisfactory responses to the TSPs. Initially, the researchers used techniques considered greedy or myopic for the development of such algorithms (Lawler et al., 1985), such as the Nearest neighbor method, the insertion method, methods for route improvement, such as *k-opt* (Laport, 1992) and Lin-Kernighan (LK) (Lin & Kernighan, 1973). It is necessary to emphasize that the LK method is a relatively old method, but it is still widely disseminated among the studies related to achieving promising solutions for the TSP since it is one of the most powerful and robust methods for that application.

A disadvantage of the heuristics lays in the difficulty of “escaping” from optimum places. This fact gave origin to another class of methodologies known as metaheuristics which have tools that allow the “escaping” from these optimum places in a way that the search can be done in more promising regions. The challenge is to produce, in minimal time, solutions that are as close as possible to the optimum solution.

It must be noted that in terms of metaheuristics, a promising approach is the Particle Swarm Optimization (PSO). The PSO algorithm was developed initially by Kennedy and Eberhart (Kennedy & Eberhart, 1995) based on the studies of the socio-biologist Edward Osborne Wilson (Wilson, 1971). The base of the development of the PSO algorithm is a hypothesis in which the exchange of information between beings of the same specie offers an evolutionary advantage.

Another recent approach is the use of concepts of Quantum mechanics (QM) (Pang, 2005) and also of Quantum Computation (QC) (Chung & Nielsen, 2000) in the design of optimization algorithms (Hogg & Portnov, 2000; Protopescu & Barhen, 2002; Coelho, 2008). The QM is a mathematical structure, or a set of rules, for the construction of physics theories. The QC was introduced by Benioff (1980) and Feynman (1982), and it is a research field in rapid growth in recent years. The concepts from QM (superposition of states; inference; delta potential energy and Schrödinger’s equation) and of the QC (quantum bits, or *qu-bits*; quantum gates and parallel processing) have been explored for the creation of new methods or even for the enhancement of the efficiency of the existing optimization methods.

The quantum world is inexplicable in terms of the classical mechanics. The previsions from the interactions of matter and light in Newton's law for particle movement, and the equations of Maxwell that control the propagation of electromagnetic fields are contradictory in experiments performed in microscopic scale. Obviously, there must be a limit line of the size of the objects so that their behavior fit in classical mechanics or in QM. In this limit line, there are objects that are approximately 100 times bigger than a hydrogen atom. The objects that are smaller than that have their behaviors described in QM, and the objects that are bigger are described by the Newtonian mechanics.

The QM is a remarkable theory. There seems to be no real doubt that much of that in physics and all in chemistry would be deductible from its postulates or laws. QM has answered many questions correctly and given a deeper view of natural processes, and it is willing to contribute even more. It is a wide theory on which is based much of our knowledge about mechanics and radiation. It is a relatively recent theory. In 1900, the matter mechanics, based on Newton's law of motion, had resisted changes for centuries. In the same way, the ondulatory theory of light, based in the electromagnetic theory, resisted with no challenges (Pohl, 1971).

It is true that QM has not yet presented a consistent description of elementary particles and of interaction fields; however the theory is already completed to be used in experiments with atoms, molecules, nuclei, radiations and solids. As a result of the expansive growth of this theory in the half of XX century, we can see its great progress in this century (Merzbacher, 1961).

The Wave Function is a mathematical tool of the quantum mechanics used to describe a physical system. It is a complex function that completely describes the state of a particle $\Psi(x, t)$. Its quadratic module is the density of probability, which is the probability of the particle to be discovered, given by equation (2) where

$$|\Psi(x, t)|^2 dx = \Psi^*(x, t)\Psi(x, t)dx \quad (2)$$

Most of the Physics theories are based on fundamental equations. For example, Newton mechanics is based on $F = ma$, classical electrodynamics is based on the equation of Maxwell, and the general relativity theory is based on Einstein's equation $G_{uv} = -8\pi GT_{uv}$. The fundamental equation of quantum mechanics is the equation of Schrödinger (Rydnik, 1979; Shankar, 1994). It is possible to write Schrödinger's equation for a m mass particle moving in a U potential in only one x dimension, as equation (3), where Ψ represents the wave function.

$$-\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2} + U\Psi = i\hbar \frac{\partial \Psi}{\partial t} \quad (3)$$

3 APPROACHED TECHNIQUES AND FUNDAMENTALS

In this section we introduce the fundamentals of the procedures of the heuristics and metaheuristics approached in this paper for solving the TSP, more specifically, Lin-Kernighan (LK); Lin-Kernighan-Helsgaun (LKH) and Quantum Particle Swarm Optimization (QPSO).

3.1 Lin-Kernighan Heuristic (LK)

An important set of heuristics developed for the TSP is made of the k -opt exchanges. In general, these are algorithms that starting from an initial feasible solution for the TSP transposes λ arcs that are in the circuit for another λ arcs outside it, keeping in mind the reduction of the total cost of the circuit at each exchange.

The first exchange mechanism known as 2 -opt was introduced by Croes (1958). In this mechanism a solution is generated through the removal of two arcs, resulting in two paths. One of the paths is inverted and then reconnected in order to form a new route. This new solution then becomes the current solution and the process is repeated until it is not possible to perform a new exchange of two arcs, with gain. Shen Lin (1965) proposes a wider neighborhood when considering all the possible solutions with the exchange of three arcs, the 3 -opt. Lin & Kernighan (1973) propose an algorithm in which the number of arcs exchanged in each step is variable. The arc exchanges (k -opt) are performed according to a criterion of gain that restricts the size of the searched neighborhood.

The LK heuristic is considered to be one of the most efficient methods in generating optimal or near-optimum solutions for the symmetric TSP. However, the design and the implementation of an algorithm based on this heuristic are not trivial. There are many decisions that must be made and most of them greatly influence in its performance. The number of operations to test all the exchanged λ increases rapidly as the number of cities increases.

In a simple implementation, the algorithm used to test the k -exchanges has the complexity of $O(n^\lambda)$. As a consequence, the values $k = 2$ and $k = 3$ are the ones used the most. In Christofides & Elion (1972), we see $k = 4$ and $k = 5$ been used. An inconvenience of this algorithm is that λ must be detailed in the beginning of the execution. And it is hard to know which value must be used for λ in order to obtain the best cost-benefit between the time of implement and the quality of solution.

Lin & Kernighan (1973) removed this inconvenience of the k -opt algorithm by introducing the k -opt variable. This algorithm can vary the value of λ during the implement, deciding at each iteration which value λ must assume. At each iteration, the algorithm searches through increasing values of λ which variation makes up the shortest route. Given a number of exchanges r , a series of tests are performed in order to verify when $r + 1$ exchange must be considered. This happens up to the point that some stop condition is met.

At each step the algorithm considers an increasing number of possible exchanges, starting in $r = 2$. These exchanges are chosen in a way that the viability of the route can be achieved in any stage of the process. If the search for a shorter route is successful then the previous route is discarded and replaced by the new route.

The LK algorithm belongs to the class of algorithms known as Local Optimization Algorithms (Johnson, 1990; Johnson & Mcgeoh, 1997). Basically, the LK algorithm consists of the following steps:

- (i) Generate a random starting tour T .
- (ii) Set $G^* = 0$. (G^* represents the best improvement made so far). Choose any node t_1 and let x_1 be one of the adjacent arcs of T to t_1 . Let $i = 1$.
- (iii) From the other endpoint t_2 of x_1 , choose y_1 to t_3 with $g_1 = |x_1| - |y_1| > 0$. If no such y_1 exists, go to (vi)(d).
- (iv) Let $i = i + 1$. Choose x_i (which currently joins t_{2i-1} to t_{2i}) and y_i as follows:
- x_i is chosen so that, if t_{2i} is joined to t_1 , the resulting configuration is a tour;
 - y_i is some available arc at the endpoint t_{2i} shared with x_i , subject to (c), (d) and (e). If no such y_i exists, go to (v);
 - In order to guarantee that the x 's and y 's are disjoint, x_i cannot be an arc previously joined (i.e., a y_j with $j < i$), and similarly y_i cannot be an arc previously broken;
 - $G_i = \sum_{j=1}^i g_j = \sum_{j=1}^i (|x_j| - |y_j|) > 0$ (Gain criterion);
 - In order to ensure that the feasibility criterion of (a) can be satisfied at $(i + 1)$, the y_i chosen must let the breaking of an x_{i+1} ;
 - Before y_i is constructed, we have to verify if closing up by join t_{2i} to t_1 will give a gain value better than the best seen previously. Let y_i^* be an arc connecting t_{2i} to t_1 , and let $g_i^* = |y_i^*| - |x_i|$. If $G_{i-1} + g_i^* > G^*$, set $G^* = G_{i-1} + g_i^*$ and let $k = i$.
- (v) Terminate the construction of x_i and y_i in steps (ii) to (iv) when either no further arcs satisfying criteria (iv)(c) to (iv)(e), or when $G_i \leq G^*$ (stopping criterion). If $G^* > 0$, take the tour T' , make $f(T') = f(T) - G^*$ and $T \leftarrow T'$ and go to step (ii).
- (vi) If $G^* = 0$, a limited backtracking is involved, as follows:
- Repeat steps (iv) and (v), choosing y_2 's in order of increasing length, as long as they satisfy the gain criterion $g_1 + g_2 > 0$;
 - If all choices of y_2 in step (iv)(b) are exhausted without gain, go to step (iv)(a) and choose another x_2 ;
 - If this also fails, go to step (iii), where the y_i 's are examined in order of increasing length;
 - If the y_i 's are also exhausted without gain, we try an alternate x_1 in step (ii);
 - If this also fails, a new t_1 is selected and we repeat step (ii).
- (vii) The procedure terminates when all n values of t_1 have been examined without gain.

In general, the LK algorithm is part of an circuit T and of an initial vehicle t_1 . In step i an exchange happens through the following sequence: arc (t_1, t_{2i}) is removed and arc (t_{2i}, t_{2i+1}) is added; arc (t_{2i+1}, t_{2i+2}) is chosen to be removed if with its removal and with the addition of arc (t_{2i+2}, t_1) a circuit is formed. Arc (t_{2i+2}, t_1) is removed if and when step $i + 1$ is executed.

The number of insertions and removals in the search process is limited by the gain criterion of LK. Basically, this criterion limits the sequence of the exchanges to those that result in positive gains (route size reduction) at each step of the sequence; at first, sequences in which the exchanges result in positive gain and others in negative gains are not taken into account, but instead, those in which the total gain is positive.

Another mechanism in this algorithm is the analysis of alternative solutions every time a new solution with zero gain is generated (step (vi)). This is done by choosing new arcs for the exchange.

3.2 Lin-Kernighan Heuristic (LKH)

Keld Helsgaun (2000) revises the original Lin-Kernighan algorithm and proposes and develops modifications that improve its performance. Efficiency is increased, firstly, by revising the heuristic rules of Lin & Kernighan in order to restrict and direct the search. Despite the natural appearance of these rules, a critical analysis will show that they have considerable deficiencies.

By analyzing the rules of the LK algorithm that are used for refining the search, Helsgaun (2000) not only saved computational time but also identified a problem that could lead the algorithm to find poor solutions for greater problems. One of the rules proposed by Lin & Kernighan (Lin & Kernighan, 1973) limits the search refining it to embrace only the closest five neighbors. In this case, if a neighbor that would provide the optimum result is not one of these five neighbors the search will not lead to an optimal result. Helsgaun (2000) uses as an example the att532 problem of the TSPLIB, by Padberg & Rinaldi (1987) in which the best neighbor is only found on the 22nd search.

Considering that fact, Helsgaun (2000) introduces a proximity measure that better reflects the possibility of getting an edge that is part of the optimum course. This measure is called α -nearness, it is based on the sensibility analysis using a minimum spanning tree *1-trees*. At least two small changes are performed in the basic motion in the general scheme. First, in a special case, the first motion of a sequence must be a *4-opt* sequential motion; the following motion must be the *2-opt* motion. In second place, the non-sequential *4-opt* motion are tested when the course cannot be improved by sequential motion.

Moreover, the modified LKH algorithm revises the basic structure of the search in other points. The first and most important point is the change of the basic *4-opt* motion to a sequential *5-opt* motion. Besides this one, computational experiences show that the *backtracking* is not so necessary in this algorithm and its removal reduces the execution time; it does not compromise the final performance of the algorithm and it extremely simplifies the execution.

Regarding the initial course, the LK algorithm performs edge exchanges many times in the same problem using different initial courses. The experiences with many execution of the LKH algorithm show that the quality of the final solutions does not depend that much on the initial solutions. However, the significant reduction in time can be achieved, as for example, by choosing initial solutions that are closer to optimum through constructive heuristics. Thus, Helsgaun also introduces in his work a simple heuristic that was used in the construction of initial solutions in his version of the LK (Lawler et al., 1985).

On the other hand, Nguyen et al. (2006) affirms that the use of the *5-opt* movement as a basic motion greatly increases the necessary computational time, although the solutions that are found are of really good quality. Besides that, the consumption of memory and necessary time for the pre-processing of the LKH algorithm are very high when compared to other implementations of LK.

3.3 QPSO (Quantum Particle Swarm Optimization)

The Particle Swarm Optimization (PSO) algorithm introduced in 1995 by James Kennedy & Russel Eberhart is based on bird's collective behaviors (Kennedy & Eberhart, 1995). The PSO algorithm is a technique of collective intelligence based in a population of solutions and random transitions. The PSO algorithm has similar characteristics to the ones of the evolutionary computation which are based on a population of solutions. However, the PSO is motivated by the simulation of social behavior and cooperation between agents instead of the survival of the individual that is more adaptable as it is in the evolutionary algorithms. In the PSO algorithm, each candidate solution (named particle) is associated to a velocity vector. The velocity vector is adjusted through an updating equation which considers the experience of the corresponding particle and the experience of the other particles in the population.

The PSO algorithm concept consists in changing the velocity of each particle in direction to the *pbest* (personal best) and *gbest* (global best) localization at each interactive step. The velocity of the search procedure is weighted by a term that is randomly generated, and that is separately linked to the localizations of the *pbest* and of the *gbest*. The procedure for implementation of the algorithm PSO is regulated by the following steps (Herrera & Coelho, 2006; Coelho & Herrera, 2006; Coelho, 2008):

- (i) Initiate randomly and with uniform distribution a particle population (matrix) with positions and speed in an space of n dimensional problem;
- (ii) For each particle evaluate the aptitude function (objective function to be minimized, in this case, the TSP optimization);
- (iii) Compare the evaluation of the aptitude function of the particle with the *pbest* of the particle. If the current value is better than *pbest*, then the *pbest* value becomes equal to the value of the particle aptitude, and the localization of the *pbest* becomes equal to the current localization in n dimensional space;
- (iv) Compare the evaluation of the aptitude function with the population best previous aptitude value. If the current value is better than the *gbest*, update the *gbest* value for the index and the current particle value;
- (v) Modify speed and position of the particle according to the equations (4) and (5), respectively, where Δt is equal to 1.

$$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot ud \cdot [p_i(t) - x_i(t)] + c_2 \cdot Ud \cdot [p_g(t) - x_i(t)] \quad (4)$$

$$x_i(t + 1) = x_i(t) + \Delta t \cdot v_i(t + 1) \quad (5)$$

- (vi) Go to step (ii) until a stop criterion is met (usually a value of error pre-defined or a maximum number of iterations).

We use the following notations: t is the iteration (generation), $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$ stores the position of the i -th particle, $v_i = [v_{i1}, v_{i2}, \dots, v_{in}]^T$ stores the velocity of the i -th particle and $p_i = [p_{i1}, p_{i2}, \dots, p_{in}]^T$ represents the position of the best fitness value of the i -th particle. The index g represents the index of the best particle among all particles of the group. The variable w is the weight of inertia; c_1 and c_2 are positive constants; ud and Ud are two functions for the generation of random numbers with uniform distribution in the intervals $[0, 1]$, respectively. The size of the population is selected according to the problem.

The particle velocities in each dimension are limited to a maximum value of velocity, V_{\max} . The V_{\max} is important because it determines the resolution in which the regions that are near the current solutions are searched. If V_{\max} is high, the PSO algorithm makes the *global* search easy, while a small V_{\max} value emphasizes the local search. The first term of equation (4) represents the term of the particle moment, where the weight of the inertia w represents the degree of the moment of the particle. The second term consists in the “cognitive” part, which represents the independent “knowledge” of the particle, and the third term consists in the “social” that represents cooperation among particles.

The constants c_1 and c_2 represent the weight of the “cognition” and “social” parts, respectively, and they influence each particle directing them to the *pbest* and the *gbest*. These parameters are usually adjusted by trial-and-error heuristics.

The Quantum PSO (QPSO) algorithm allows the particles to move following the rules defined by the quantum mechanics instead of the classical Newtonian random movement (Sun et al., 2004a, 2004b). In the quantum model of the PSO, the state of each particle is represented by a wave function $\Psi(\bar{x}, t)$ instead of the velocity position as it is in the conventional model. The dynamic behavior of the particle is widely divergent of the PSO traditional behavior, where exact values and velocities and positions cannot be determined simultaneously. It is possible to learn the probability of a particle to be in a certain position through the probability of its density function $|\Psi(\bar{x}, t)|^2$ which depends on the potential field in which the particle is.

In Sun et al. (2004a) and Sung et al. (2004b), the wave function of the particle can be defined by equation (6) where

$$\Psi(x) = \frac{1}{\sqrt{L}} \cdot e^{\frac{-\|p-x\|}{L}} \quad (6)$$

and the density of probability is given by the following expression (7).

$$Q(x) = |\Psi(x)|^2 = \frac{1}{L} \cdot e^{\frac{-2\|p-x\|}{L}} \quad (7)$$

The L parameter depends on the intensity of energy in the potential field, which defines the scope of search of the particle and can be called the Creativity or Imagination of the particle (Sun et al., 2004b).

In the quantum inspired PSO model, the search space and the solution space is different in quality. The wave function or the probability function describes the state of the particle in a quantum search space, and it does not provide any information of the position of the particle, what is mandatory to calculate the cost function.

In this context the transformation of a state among these two spaces is essential. Concerning the quantum mechanics, the transformation of a quantum state to a classical state defined in the conventional mechanics is known as collapse. In nature, this is the measure of the position of the particle. The differences between the conventional PSO model and the quantum inspired model are shown in Figure 1 (Sun et al., 2004b).

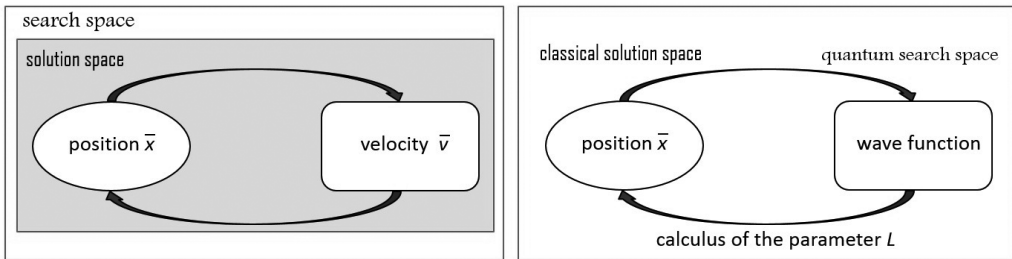


Figure 1 – PSO and QPSO search space.

Due to the quantum nature of the equations, the admeasurements using conventional computers must use the Monte Carlo stochastic simulation method (MMC). The particle position can be defined by equation (8).

$$x(t) = p \pm \frac{L}{2} \ln \left(\frac{1}{n} \right). \tag{8}$$

In Sun et al. (2004a) the L parameter is calculates as in (9) where:

$$L(t + 1) = 2 \cdot \alpha \cdot \|p - x(t)\|. \tag{9}$$

The QPSO interactive equation is defined as

$$x(t + 1) = p \pm \alpha \cdot \|p - x(t)\| \cdot \ln \left(\frac{1}{u} \right), \text{ with } u = rand(0, 1) \tag{10}$$

that substitutes equation (5) of the conventional PSO algorithm model. Concerning knowledge evolution of a social organism, there are two types of thoughts related to the way individuals in a population acquire knowledge. The first one is through the $pbest$, the best value found by the individual, and the second it the $gbest$, the best solution found by the swarm (population). Each particle does the search in the direction of the current position of the individual to the p point that is located between the $pbest$ and the $gbest$. The p point is known as LIP (Learning Inclination Point) of the individual. The learning tendency of each individual leads their search to their LIP neighborhood, which is determined by the $pbest$ and the $gbest$. The coefficient α is called Contraction-Expansion term, which can be tuned to control the convergence speed of the algorithms.

In the QPSO algorithm, each particle records its *pbest* and compares it to all the other particles of the population in order to get the *gbest* at each iteration. In order to execute the next step, the L parameter is calculated. The L parameter is considered to be the Creativity or Imagination of the particle, and therefore it is characterized as the scope of search of the knowledge of the particle. The greater the value of L , the easier the particle acquires new knowledge. In the QPSO the Creativity factor of the particle is calculated as the difference between the current position of the particle and its LIP, as shown in equation (9).

In Shi & Eberhart (1999), the position of the best mean (*mbest*) is introduced to the PSO, and in Sun et al. (2005a) in the QPSO, the *mbest* is defined as the mean of the *pbest* of all the particles in the swarm (population), given by expression (11).

$$mbest = \frac{1}{M} \sum_{i=1}^M p_i = \left(\frac{1}{M} \sum_{i=1}^M p_{i1}, \frac{1}{M} \sum_{i=1}^M p_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M p_{id} \right) \quad (11)$$

where M is the size of the population and p_i is the *pbest* of i th particle. Thus, equations (9) and (10) can be redefined as (12) and (13), respectively.

$$L(t+1) = 2 \cdot \alpha \cdot \|mbest - x(t)\| \quad (12)$$

$$x(t+1) = P \pm \alpha \cdot \|mbest - x(t)\| \cdot \ln\left(\frac{1}{u}\right), \quad \text{with } u = rand(0, 1) \quad (13)$$

The pseudocode for the QPSO algorithm is described in the following Algorithm 1.

The quantum model of the QPSO shows some advantages in relation to the traditional model. Some peculiarities of the QPSO according to Sun et al. (2004a) can be cited. They are the following: quantum systems are complex systems and nonlinear based on the Principle of Superposition of States, in other words, the quantum models have a lot more states than the conventional model; quantum systems are uncertain systems, thus they are very different from the stochastic classical system. Before measurement the particle can be in any state with certain probability and no predetermined final course.

The QPSO algorithm approached up to this point works for continuous problems and not for discrete ones, as the TSP. Some alterations must be implemented in order for this algorithm to be used with discrete problems.

Consider an initial M population of size four and dimension four represented by the following matrix:

$$M = \begin{bmatrix} 1.02 & 3.56 & -0.16 & 4.5 \\ 1.14 & -0.10 & 5.27 & 1.65 \\ 1.63 & 1.52 & 0.48 & -1.24 \\ 1.99 & 0.97 & -1.82 & 2.24 \end{bmatrix}$$

where each line of matrix M represents a possible solution for a continuous problem, for example, the minimization of the sphere function $f(x) = \sum_{i=1}^n x_i^2$. In order to know the *gbest* of

Algorithm 1 – QPSO Pseudocode.

```

Initial Population: Random  $x_i$ 
Do
  For  $i = 1$  until population size  $M$ 
    If  $f(x_i) < f(p_i)$  then  $p_i = x_i$ 
     $p_g = \min(p_i)$ 
    Calculate  $mbest$  using equation (11)
    For  $d = 1$  until dimension  $D$ 
       $f_{i1} = rand(0, 1)$ ,  $f_{i2} = rand(0, 1)$ 
       $p = \frac{f_{i1} \cdot p_{id} + f_{i2} \cdot p_{gd}}{f_{i1} + f_{i2}}$ 
       $u = rand(0, 1)$ 
      If  $rand(0, 1) > 0.5$ 
         $x_{id} = p - \alpha \cdot |mbest_d - x_{id}| \cdot \ln\left(\frac{1}{u}\right)$ 
      Else
         $x_{id} = p + \alpha \cdot |mbest_d - x_{id}| \cdot \ln\left(\frac{1}{u}\right)$ 
    Until stopping criterion is met

```

this initial population, one must calculate the $pbest$ of each particle (line) and then verify which is the smallest one. In other words, the particle that shows the smaller $pbest$ also has the smaller $gbest$. One must note that depending on the cost function associated to the problem the dimension of the population can be fixed, or, as in the above example, varied. The QPSO can be applied in this case.

Now we have the following context for the TSP. Once matrix M is generated by a uniform distribution, the question is, how can we calculate the objective function associated to the TSP, defined in equation (1), once the cities (vertices of the graph) are numbers that belong to the positive integers?

As a solution we must discretize the continuous values of matrix M in order to calculate the objective or cost function. Matrix M is not modified and its values persist in the execution of the algorithm of the QSPO. The discretization is performed only at the moment the objective function is calculated (Eq. (1)) for a determined solution. Thus, each line of matrix M , a possible solution for problems with four cities, must be discretized.

Therefore, the following rule is used: the lower value of the line will represent the first city; the second lower value, the second city, and so on. This type of approach has been proposed in the literature in Tasgeriten & Liang (2003). Some works about the application of PSO approaches for combinatory problems have been introduced in recent literature, such as Wang et al., 2003;

Pang et al., 2004a; Pang et al., 2004b; Machado & Lopes, 2005; Lopes & Coelho, 2005, but none of them using QPSO.

Thus, for the first line of matrix M we have:

$$\begin{bmatrix} 1.02 & 3.56 & -0.16 & 4.5 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 3 & 1 & 4 \end{bmatrix}$$

where $\begin{bmatrix} 2 & 3 & 1 & 4 \end{bmatrix}$ represent a solution for the TSP of four cities. The total discretized matrix M is shown below:

$$M = \begin{bmatrix} 1.02 & 3.56 & -0.16 & 4.5 \\ 1.14 & -0.10 & 5.27 & 1.65 \\ 1.63 & 1.52 & 0.48 & -1.24 \\ 1.99 & 0.97 & -1.82 & 2.24 \end{bmatrix} \Rightarrow M_{discrete} = \begin{bmatrix} 2 & 3 & 1 & 4 \\ 2 & 1 & 4 & 3 \\ 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 4 \end{bmatrix} \begin{matrix} \textit{tour}(1) \\ \textit{tour}(2) \\ \vdots \end{matrix}$$

In this case, there is an evident problem that derives of the use of this simple approach which is clear when there are repeated values in matrix M . In greater problems (many cities) it is possible that some repeated columns may exist in a certain line of matrix M ; for example, if the last line of matrix M given by [1.99; 1.82; 2.24; 1.99] will be transformed to [2; 1; 4; 3]. In this case, the 1st number 1.99 (1st position) has priority in relation to the 2nd number 1.99 (4th position) due the minor position in the last line of matrix M .

This fact becomes normal with instances of the TSP with a number of increased cities. The duplicate of the values may not affect the TSP that has an increased number of cities. The duplicate of values may not affect the execution of some discrete problems of the CO, however, in the case of the TSP, repeated vertices in the solution are not allowed.

In order to solve this problem the algorithm represented by the pseudocode in the following Algorithm 2 is used.

4 COMPUTATIONAL IMPLEMENTATION AND ANALYSIS OF RESULTS

In this section we present the results from the experiments using the heuristics of LKH enhancement and the metaheuristic QPSO previously discussed in section 3 and also a statistics analysis of the same.

The executed algorithms were applied to the instances of the repository TSPLIB. Small, medium and large sized instances were selected to test the optimization approaches selected for the TSP.

For each instance of the problem the algorithm was executed 30 times using different seeds in each of them. The execution form used was the following:

- (i) QPSO generates a random initial population;
- (ii) The best individual (*tour*) of this initial population is recorded in a archive of extension *.ini* (Used afterwards by the Rand+LKH);

- (iii) QPSO is executed until it finds the best global for the population randomly generated;
- (iv) The best global (*tour*) is recorded in an archive with extension *.final*;
- (v) The LKH is executed using as the initial tour the archive (**.ini*) (Rand+LK approach);
- (vi) The LKH is executed using as the initial tour the archive (*.final*) (QPSO+LK Approach);
- (vii) The LKH is executed with default parameters (pure LKH).

The size of the populations was fixed in 100 particles and the stop criterion was the optimum value mentioned in the TSPLIB for the approached problem. In the case of the QPSO without LKH, the stopping criterion was a fixed iterations number previously defined as 1000.

In QPSO, Contraction-Expansion Coefficient α is a vital parameter to the convergence of the individual particle in QPSO, and therefore exerts significant influence on convergence of the algorithm. Mathematically, there are many forms of convergence of stochastic process, and different forms of convergence have different conditions that the parameter must satisfy. An efficient method is linear-decreasing method that decreasing the value of α linearly as the algorithm is running. In this paper, α is adopted with value 1.0 at the beginning of the search to 0.5 at the end of the search for all optimization problems (Sun et al., 2005b).

4.1 Results for the Symmetric TSP

Table 1 shows the results for the algorithms: (i) QPSO+LKH, (ii) Rand +LKH, and (iii) LKH, for 10 test problems from the TSPLIB (Reinelt, 1994). The notation “%” was used in this table to represent how much a value for the test problem is distant from the optimum value in percentage.

Note by the results in Table 1 that for the problem swiss42 (Reinelt, 1994), the algorithms QPSO+LKH, Rand+LKH and LKH showed similar performance, including in the statistical analysis by obtaining the best value (optimum value) for the objective function of 1273. Regarding the problem Gr229, the algorithm Rand+LKH did not achieved the optimum value of 134602, instead, 0.010% of this value. However, the QPSO+LKH and the LKH achieved optimum value. The LKH was in mean slightly superior to the QPSO+LKH.

For the problem pbc442 the QPSO+LKH was the fastest algorithm. However the three optimization approaches achieved optimum value for the objective function which is of 50778. Based on the results of the simulation for the problem Gr666 showed in Table 1 note that the lower standard deviation was achieved by QPSO+LKH but the mean of the results of the objective function for the QPSO+LKH and LKH was identical.

As for the problem dsj1000, all the tested algorithms achieved optimum results. However, in terms of convergence, the LKH showed the best mean of results for the objective function. Concerning the problem pr1002, the optimization algorithms achieved optimum value for the TSP, but the Rand+LKH was the fastest algorithm. For the pcb1173, the Rand+LKH was the optimization approach with better mean of values achieved for objective function.

Table 1 – Results of the optimization of the instances for the symmetric TSP.

Instance (optimum)	Method	Minimum (%)	Mean (%)	Maximum (%)	Median (%)	Standard deviation time(s)
swiss42 (1273)	QPSO+LKH	1273	1273.000	1273	1273.0	0.00
	Rand+LKH	1273	1273.000	1273	1273.0	0.00
	LKH	1273	1273.000	1273	1273.0	0.00
Gr229 (134602)	QPSO+LKH	134602	134615.533	134616	134616.0	2.56
	Rand+LKH	134616	134616.000	134616	134616.0	1.8
	LKH	134602	134613.200	134616	134616.0	1.7
pcb442 (50778)	QPSO+LKH	50778	50778.233	50785	50778.0	5.70
	Rand+LKH	50778	50778.233	50785	50778.0	0.1
	LKH	50778	50778.233	50785	50778.0	0.3
Gr666 (294358)	QPSO+LKH	294358	294444.667	294476	294476.0	1.28
	Rand+LKH	294358	294426.833	294476	294476.0	1.28
	LKH	294358	294426.833	294476	294476.0	0.8
dsj1000 (18659688)	QPSO+LKH	18660188	18664570.200	18681851	18660188.0	52.63
	Rand+LKH	18660188	18666030.933	18682099	18660188.0	15.9
	LKH	18660188	18664537.133	18681851	18660188.0	11.8
Pr1002 (259045)	QPSO+LKH	259045	259045.000	259045	259045.0	60.76
	Rand+LKH	259045	259045.000	259045	259045.0	6.6
	LKH	259045	259045.000	259045	259045.0	34
pcb1173 (56892)	QPSO+LKH	56892	56893.000	56897	56892.0	8944.22
	Rand+LKH	56892	56892.333	56897	56892.0	0.003
	LKH	56892	56893.000	56897	56892.0	9914.04
dl1291 (50801)	QPSO+LKH	50801	50849.750	50886	50868.5	28.16
	Rand+LKH	50801	50843.500	50886	50843.5	0.00
	LKH	50801	50830.100	50886	50801.0	2.6
ui1817 (50201)	QPSO+LKH	57201	57242.833	57313	57243.0	0.00
	Rand+LKH	57225	57246.250	57272	57245.5	2.8
	LKH	57201	57237.083	57272	57236.5	2.07
Fl13795 (28772)	QPSO+LKH	28772	28779.231	28813	28772.0	8.3
	Rand+LKH	28772	28788.692	28813	28785.0	1.29
	LKH	28772	28808.846	28881	28813.0	2.07
						9.2
						40.4
						40.2
						79.87
						75.5
						93.1
						117
						1310.3
						1787
						2367.3

For the problems d1291 and u1817, the LKH was the method with the best means, remembering that the QPSO+LKH achieved optimum value for at least 30 of the experiments. Note that the Rand+LKH achieved optimum for the d1291, but the best result is 0.042% to reach optimum for problem u1817.

For the large scale symmetric TSP tested in this paper, note that the QPSO+LKH was better and faster than the Rand+LKH and the LKH. Figure 2 shows the best route found for the test problem pcb1173.

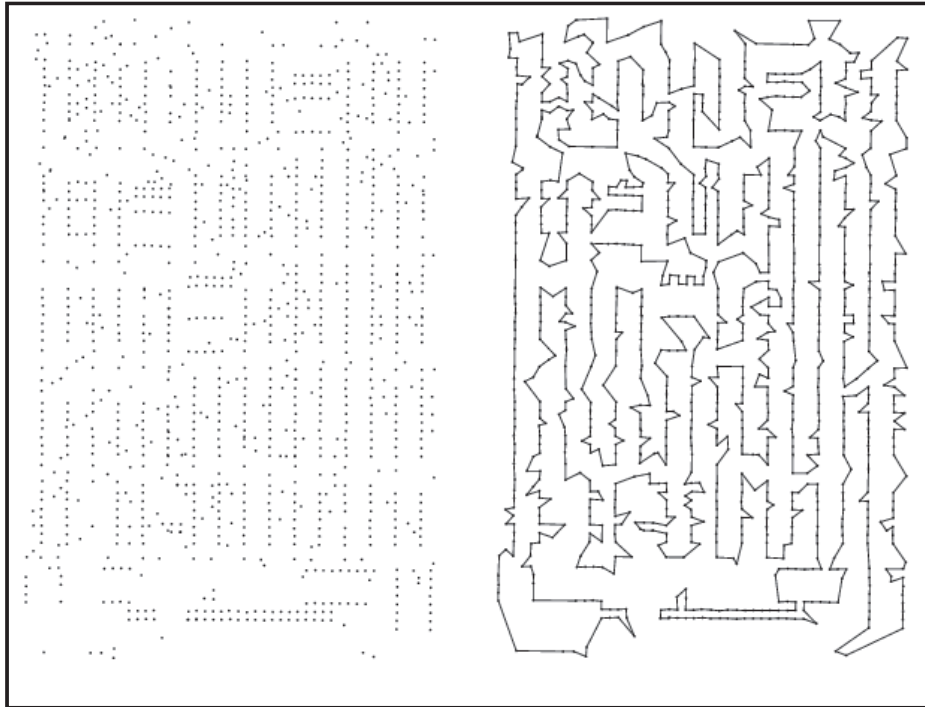


Figure 2 – Example of the best route found for the problem pcb1173.

4.2 Asymmetric Problems

Table 2 shows the results for the algorithms (i) QPSO+LKH, (ii) Rand+LKH and (iii) LKH for the four test problems of the asymmetric TSP of the TSPLIB. As in Table 1, Table 2 used the notation “%” in order to represent how much percent is the achieved value of the test problem distant from the optimum value.

Note that by the results on Table 2, all the algorithms achieved optimum value for the problems ftv38 and rg443, but, the LKH was the fastest. For ftv323, the Rand+LKH was the slowest among the tested algorithms.

Table 2 – Results of optimization of the instances for the asymmetric TSP.

Instance (optimum)	Method	Minimum (%)		Mean (%)		Maximum (%)		Median (%)		Standard deviation time(s)	
ftv38	QPSO+LKH	1530	0.00	1532.00	0.13	1530.53	0.03	1530.00	0.00	0.90	0.1
(1530)	Rand+LKH	1530	0.00	1532.00	0.13	1530.33	0.02	1530.00	0.00	0.76	0.1
	LKH	1530	0.00	1532.00	0.13	1530.20	0.01	1530.00	0.00	0.61	≈ 0
ftv170	QPSO+LKH	2755	0.00	2755.00	0.00	2755.00	0.00	2755.00	0.00	0.00	0.1
(2755)	Rand+LKH	2755	0.00	2755.00	0.00	2755.00	0.00	2755.00	0.00	0.00	0.4
	LKH	2755	0.00	2755.00	0.00	2755.00	0.00	2755.00	0.00	0.00	0.1
rg323	QPSO+LKH	1326	0.00	1327.73	0.13	1328.00	0.15	1328.00	0.15	0.70	29.1
(1326)	Rand+LKH	1326	0.00	1327.33	0.10	1328.00	0.15	1328.00	0.15	0.98	24.9
	LKH	1326	0.00	1327.07	0.08	1328.00	0.15	1328.00	0.15	1.03	20.2
rg443	QPSO+LKH	2720	0.00	2720.00	0.00	2720.00	0.00	2720.00	0.00	0.00	163
(2720)	Rand+LKH	2720	0.00	2720.00	0.00	2720.00	0.00	2720.00	0.00	0.00	164
	LKH	2720	0.00	2720.00	0.00	2720.00	0.00	2720.00	0.00	0.00	55

5 CONCLUSIONS AND FUTURE WORKS

The troubleshooting of the CO as, for example, the TSP, can be resolved by using recent approaches such as particle swarm concepts and quantum mechanics. The metaheuristic QPSO is an optimization method based on the simulation of the social interaction among individuals in a population. Each element in it moves in a hyperspace, attracted by positions (promising solutions).

The LK heuristic is considered one of the most efficient methods for generating optimum or near-optimum solutions for the symmetric TSP. However, the design and the execution of an algorithm based on this heuristic are not trivial. There are many possibilities for decision making and most of them have a significant influence on the performance (Helsgaun, 2000).

As the Lin-Kernighan heuristic, the QPSO metaheuristic shows many parameters of settings and implementations that directly affects the performance of the proposed algorithm. The algorithm shows promising results for the greater instances ($n > 1000$), range in which the LKH does not produces such efficient results, even though it did not show satisfactory results for small instances of the problem ($n < 1000$) (Nguyen et al., 2007).

New investigations can be done by varying the parameters of the QPSO algorithm and so adapting them to each of the instances to be tested. The size of the population, number of iterations and obviously, the LIP can vary. This probably would lead to an improvement of the performance since each problem, even if the objective function is the same, show variations of behavior. There might be, for example, instances where clustering works well, as it is in the case of the algorithm proposed by Neto (1999).

REFERENCES

- [1] ALBAYRAK M & ALLAHVERDI N. 2011. Development a New Mutation Operator to Solve the Traveling Salesman Problem by Aid of Genetic Algorithms. *Expert Systems with Applications*, **38**: 1313–1320.
- [2] BAGHERI A, PEYHANI HM & AKBARI M. 2014. Financial Forecasting Using ANFIS Networks with Quantum-behaved Particle Swarm Optimization. *Expert Systems with Applications*, **41**: 6235–6250.
- [3] BENIOFF P. 1980. The Computer as a Physical System: a Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines. *Journal of Statistical Physics*, **22**: 563–591.
- [4] CHEN SM & CHIEN CY. 2011. Parallelized Genetic Ant Colony Systems for Solving the Traveling Salesman Problem. *Expert Systems with Applications*, **38**: 3873–3883.
- [5] CHEN SM & CHIEN CY. 2011. Solving the Traveling Salesman Problem based on the Genetic Simulated Annealing and Colony System with Particle Swarm Optimization Techniques. *Expert Systems with Applications*, **38**: 14439–14450.
- [6] CHRISTOFIDES N & EILON S. 1972. Algorithms for Large-scale Traveling Salesman Problems. *Operational Research*, **23**: 511–518.
- [7] CHUANG I & NIELSEN M. 2000. Quantum Computation and Quantum Information. *Cambridge University Press*, Cambridge, England.
- [8] COELHO LS & HERRERA BM. 2006. Fuzzy Modeling Using Chaotic Particle Swarm Approaches Applied to a Yo-yo Motion System. *Proceedings of IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, pp. 10508–10513.
- [9] COELHO LS. 2008. A Quantum Particle Swarm Optimizer with Chaotic Mutation Operator. *Chaos, Solutions and Fractals*, **37**: 1409–1418.
- [10] CROES G. 1958. A Method for Solving Traveling Salesman Problems. *Operations Research*, **6**: 791–8112.
- [11] DONG G, GUO WW & TICKLE K. 2012. Solving the Traveling Salesman Problem using Cooperative Genetic Ant Systems. *Expert Systems with Applications*, **39**: 5006–5011.
- [12] FANG W, SUN J, DING Y, WU X & XU W. 2010. A Review of Quantum-behaved Particle Swarm Optimization. *IETE Technical Review*, **27**: 336–348.
- [13] FEYMANN RP. 1982. Simulating Physics with Computers. *International Journal of Theoretical Physics*, **21**: 467–488.
- [14] GROVER LK. 1996. A Fast Quantum Mechanical Algorithm for Database Search, Proceedings of the 28th ACM Symposium on Theory of Computing (STOC), Philadelphia, PA, USA, pp. 212–219.
- [15] HAMED HNA, KASABOV NK & SHAMSUDDIN SM. 2011. Quantum-Inspired Particle Swarm Optimization for Feature Selection and Parameter Optimization in evolving Spiling Networks for Classification Tasks. In: *Evolucionary Algorithms* (pp. 133–148). Croatia: Intech, 2011.
- [16] HELSGAUN K. 2000. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research*, **126**: 106–130.

- [17] HERRERA BM & COELHO LS. 2006. Nonlinear Identification of a Yo-yo System Using Fuzzy Model and Fast Particle Swarm Optimization. *Applied Soft Computing Technologies: The Challenge of Complexity*, A. Abraham, B. de Baets, M. Köppen, B. Nickolay (editors), Springer, London, UK, pp. 302–316.
- [18] HOFFMANN KL. 2000. Combinatorial Optimization: Current Successes and Directions for the Future. *Journal of Computational and Applied Mathematics*, **124**: 341–360.
- [19] HOGG T & PORTNOV DS. 2000. Quantum Optimization. *Information Sciences*, **128**: 181–197.
- [20] HOSSEINNEZHAD V, RAFIEE M, AHMADIAN M & AMELI MT. 2014. Species-based Quantum Particle Swarm Optimization for Economic Load Dispatch. *International Journal of Electrical Power & Energy Systems*, **63**: 311–322.
- [21] ISHIBUCHI H & MURATA T. 1998. Multi-objective Genetic Local Search Algorithm and its Application to Flowshop Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, **28**: 392–403.
- [22] JASZKIEWICZ A. 2002. Genetic Local Search for Multi-objective Combinatorial Optimization. *European Journal of Operational Research*, **137**: 50–71.
- [23] JAU Y-M, SU K-L, WU C-J & JENG J-T. 2013. Modified Quantum-behaved Particle Swarm Optimization for Parameters Estimation of Generalized Nonlinear Multi-regressions Model Based on Choquet integral with Outliers. *Applied Mathematics and Computation*, **221**: 282–295.
- [24] JOHNSON DS. 1990. Local Optimization and the Traveling Salesman Problem. *Lecture Notes in Computer Science*, **442**: 446–461.
- [25] JOHNSON DS & MCGEOCH LA. 1997. The Traveling Salesman Problem: A Case Study in Local Optimization in: AARTS EHL & LENSTRA JK (eds.), *Local Search in Combinatorial Optimization*, John Wiley & Sons, INC, New York, NY, USA.
- [26] KENNEDY J & EBERHART R. 1995. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942–1948.
- [27] LAPORT G. 1992. The Traveling Salesman Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, **59**: 231–247.
- [28] LAWLER EL, LENSTRA JK & SHMOYS DB. 1985. *The traveling salesman problem: A guided tour of combinatorial optimization*. Chichester: Wiley Series in Discrete Mathematics & Optimization.
- [29] LI Y, JIAO L, SHANG R & STOLKIN R. 2015. Dynamic-context Cooperative Quantum-behaved Particle Swarm Optimization Based on Multilevel Thresholding Applied to Medical Image Segmentation. *Information Sciences*, **294**: 408–422.
- [30] LIN L, GUO F, XIE X & LUO B. 2015. Novel Adaptive Hybrid Rule Network Based on TS Fuzzy Rules Using an Improved Quantum-behaved Particle Swarm Optimization. *Neurocomputing, Part B*, **149**: 1003–1013.
- [31] LIN S. 1965. Computer Solutions for the Traveling Salesman Problem. *Bell Systems Technology Journal*, **44**: 2245–2269.
- [32] LIN S & KERNIGHAN BW. 1973. An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Research*, **21**: 498–516.

- [33] LIU F & ZENG G. 2009. Study of Genetic Algorithm with Reinforcement Learning to Solve the TSP. *Expert Systems with Applications*, **36**: 6995–7001.
- [34] LOPES HS & COELHO LS. 2005. Particle Swarm Optimization with Fast Local Search for the Blind Traveling Salesman Problem. *Proceedings of 5th International Conference on Hybrid Intelligent Systems*, Rio de Janeiro, RJ, pp. 245–250.
- [35] LOURENÇO HR, PAIXÃO JP & PORTUGAL R. 2001. Multiobjective Metaheuristics for the Bus-driver Scheduling Problem. *Transportation Science*, **35**: 331–343.
- [36] MACHADO TR & LOPES HS. 2005. A Hybrid Particle Swarm Optimization Model for the Traveling Salesman Problem. H. Ribeiro, R. F. Albrecht, A. Dobnikar, *Natural Computing Algorithms*, Springer, New York, NY, USA, pp. 255–258.
- [37] MANJU A & NIGAM MJ. 2014. Applications of Quantum Inspired Computational Intelligence: a survey, *Artificial Intelligence Review*, **49**: 79–156.
- [38] MARIANI VC, DUCK ARK, GUERRA FA, COELHO LS & RAO RV. 2012. A Chaotic Quantum-behaved Particle Swarm Approach Applied to Optimization of Heat Exchangers. *Applied Thermal Engineering*, **42**: 119–128.
- [39] MERZBACHER E. 1961. Quantum Mechanics. *John Wiley & Sons*, New York, NY, USA.
- [40] MISEVICIUS A, SMOLINSKAS J & TOMKEVICIUS A. 2005. Iterated Tabu Search for the Traveling Salesman Problem: new results. *Information Technology and Control*, **34**: 327–337.
- [41] MOSHEIOV G. 1994. The Traveling Salesman Problem with Pick-up and Delivery. *European Journal of Operational Research*, **79**: 299–310.
- [42] NAGATA Y & SOLER D. 2012. A New Genetic Algorithm for the Asymmetric Traveling Salesman Problem. *Expert Systems with Applications*, **39**: 8947–8953.
- [43] NEMHAUSER GL & WOLSEY AL. 1988. Integer and Combinatorial Optimization. *John Wiley & Sons*, New York, NY, USA.
- [44] NETO DM. 1999. Efficient Cluster Compensation for Lin-Kernighan Heuristics. *PhD Thesis*, Department of Computer Science, University of Toronto, Canada.
- [45] NGUYEN HD, YOSHIHARA I & YAMAMORI M. 2007. Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems. *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics*, **37**: 92–99.
- [46] NGUYEN HD, YOSHIHARA I, YAMAMORIK & YASUNAGA M. 2006. Lin-Kernighan Variant. Link: <http://public.research.att.com/~dsj/chtsp/nguyen.txt>
- [47] PADBERG MW & RINALDI G. 1987. Optimization of a 532-city Symmetric Traveling Salesman Problem by Branch and Cut. *Operations Research Letters*, **6**: 1–7.
- [48] PANG WJ, WANG KP, ZHOU CG & DONG LJ. 2004a. Fuzzy Discrete Particle Swarm Optimization for Solving Traveling Salesman Problem. *Proceedings of 4th International Conference on Computer and Information Technology*, Washington, DC, USA, pp. 796–800.
- [49] PANG W, WANG KP, ZHOU CG, DONG LJ, LIU M, ZHANG HY & WANG JY. 2004b. Modified Particle Swarm Optimization Based on Space Transformation for Solving Traveling Salesman Problem. *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, **4**: 2342–2346.

- [50] PANG XF. 2005. Quantum Mechanics in Nonlinear Systems. *World Scientific Publishing Company*, River Edge, NJ, USA.
- [51] PAPADIMITRIOU CH & STEIGLITZ K. 1982. Combinatorial Optimization – Algorithms and Complexity. *Dover Publications*, New York, NY, SA.
- [52] POP PC, KARA I & MARC AH. 2012. New Mathematical Models of the Generalized Vehicle Routing Problem and Extensions. *Applied Mathematical Modelling*, **36**: 97–107.
- [53] PURIS A, BELLO R & HERRERA F. 2010. Analysis of the Efficacy of a Two-Stage Methodology for Ant Colony Optimization: Case of Study with TSP and QAP. *Expert Systems with Applications*, **37**: 5443–5453.
- [54] REINELT G. 1994. The Traveling Salesman: Computational Solutions for TSP Applications. *Lecture Notes in Computer Science*, 840.
- [55] ROBATI A, BARANI GA, POUR HNA, FADAAE MJ & ANARAKI JRP. 2012. Balanced Fuzzy Particle Swarm Optimization. *Applied Mathematical Modelling*, **36**: 2169–2177.
- [56] RYDNIK V. 1979. ABC's of Quantum Mechanics. *Peace Publishers*, Moscow, U.R.S.S.
- [57] SAUER JG, COELHO LS, MARIANI VC, MOURELLE LM & NEDJAH N. 2008. A Discrete Differential Evolution Approach with Local Search for Traveling Salesman Problems. *Proceedings of the 7th IEEE International Conference on Cybernetic Intelligent Systems*, London, UK.
- [58] SHANKAR R. 1994. Principles of Quantum Mechanics, 2nd Edition, Plenum Press.
- [59] SHOR PW. 1994. Algorithms for Quantum Computation: Discrete Logarithms and Factoring, *Proceedings of the 35th Annual Symposium Foundations of Computer Science*, Santa Fe, USA, pp. 124–134.
- [60] SHI Y, EBERHART R. 1999. Empirical study of particle swarm optimization. *Proceedings of Congress on Evolutionary Computation*, Washington, DC, USA, pp. 1945–1950.
- [61] SILVA CA, SOUSA JMC, RUNKLER TA & SÁ DA COSTA JMG. 2009. Distributed Supply Chain Management using Ant Colony Optimization. *European Journal of Operational Research*, **199**: 349–358.
- [62] SOSA NGM, GALVÃO RD & GANDELMAN DA. 2007. Algoritmo de Busca Dispersa aplicado ao Problema Clássico de Roteamento de Veículos. *Revista Pesquisa Operacional*, **27**(2): 293–310.
- [63] STEINER MTA, ZAMBONI LVS, COSTA DMB, CARNIERI C & SILVA AL. 2000. O Problema de Roteamento no Transporte Escolar. *Revista Pesquisa Operacional*, **20**(1): 83–98.
- [64] SUN J, FANG W, WU X, XIE Z & XU W. 2011. QoS Multicast Routing Using a Quantum-behaved Particle Swarm Optimization Algorithm. *Engineering Applications of Artificial Intelligence*, **24**: 123–131.
- [65] SUN J, FENG B & XU W. 2004a. Particle Swarm Optimization with Particles Having Quantum Behavior. *Proceedings of Congress on Evolutionary Computation*, Portland, Oregon, USA, pp. 325–331.
- [66] SUN J, FENG B & XU W. 2004b. A Global Search Strategy of Quantum-Behaved Particle Swarm Optimization. *Proceedings of IEEE Congress on Cybernetics and Intelligent Systems*, Singapore, pp. 111–116.

- [67] SUN J, XU W & FENG B. 2005a. Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Big Island, HI, USA, pp. 3049–3054.
- [68] SUN J, XU W & LIU J. 2005b. Parameter Selection of Quantum-Behaved Particle Swarm Optimization, International Conference Advances in Natural Computation (ICNC), Changsha, China, Lecture Notes on Computer Science (LNCS) 3612, pp. 543–552.
- [69] SUN J, XU W, PALADE V, FANG W, LAI C-H & XU W. 2012. Convergence Analysis and Improvements of Quantum-behaved Particle Swarm Optimization. *Information Sciences*, **193**: 81–103.
- [70] TASGERITEN MF & LIANG YC. 2003. A Binary Particle Swarm Optimization for Lot Sizing Problem. *Journal of Economic and Social Research*, **5**: 1–20.
- [71] WANG KP, HUANG L, ZHOU CG, PANG CC & WEI P. 2003. Particle Swarm Optimization for Traveling Salesman Problem. *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, Xi'an, China, pp. 1583–1585.
- [72] WANG Y, FENG XY, HUANG YX, PU DB, LIANG CY & ZHOU WG. 2007. A novel quantum swarm evolutionary algorithm and its applications. *Neurocomputing*, **70**: 633–640.
- [73] WILSON EO. 1971. *The Insect Societies*. *Belknap Press of Harvard University Press*, Cambridge, MA.
- [74] ZANG B, QI H, SUN S-C, EUAN L-M & TAN H-P. 2015. Solving Inverse Problems of Radiative Heat Transfer and Phase Change in Semitransparent Medium by using Improved Quantum Particle Swarm Optimization. *International Journal of Heat and Mass Transfer*, **85**: 300–310.