A Multi-Core Iterated Local Search for the Traveling Salesman Problem using OpenMP

Isaac Kosloski Oliveira[1], **Bianca de Almeida Dantas**[1], **Graziela Santos Araújo**[1]

[1]College of Computing – Federal University Mato Grosso do Sul (UFMS)
Campo Grande – MS – Brazil

{isaac.kosloski,bianca.dantas,graziela.araujo}@ufms.br

Abstract. *This paper presents the implementation and evaluation of a sequential and a parallel metaheuristic algorithm, the Iterated Local Search (ILS), to solve the Traveling Salesman Problem (TSP). The algorithm iteratively applies a local search strategy to improve solutions. The study compares accuracy, efficiency, and speedup on TSP benchmarks, demonstrating that the algorithm produces reasonable quality solutions with faster computational speed in the parallel version. This scalability indicates that this approach may be suitable for real-world TSP applications.*

Resumo. *Este artigo implementa e avalia versões sequenciais e paralelas de um algoritmo meta-heurístico, o Iterated Local Search (ILS), para resolver o Problema do Caixeiro Viajante (TSP). O algoritmo aplica iterativamente busca local para melhorar as soluções. O estudo compara a precisão, eficiência e aceleração em benchmarks do TSP, demonstrando que o algoritmo produz soluções de boa qualidade com maior velocidade computacional na versão paralela. Essa escalabilidade indica que essa abordagem pode ser adequada para aplicações do TSP no mundo real.*

1. Introduction

One of the most prominent combinatorial optimization problems is the *traveling salesman problem* (TSP). This problem is usually modeled as a complete graph with n vertices, where the set of the vertices represents a group of cities, and the set of the arcs typifies a group of roads interconnecting the cities; the main target is for the salesman to make a tour, visiting each city exactly once and finishing at the city he has started from [Thomas H. Cormen and Stein 2009]. Some real-world applications can be found and implemented as a TSP or a variation. Examples are the drilling problem for printed circuit boards (PCBs), overhauling gas turbine engines, determining a sequence of operations to control a robot, vehicle routing problems, scheduling problems, and mask plotting in printed circuit boards (PCBs) production, among others [Reinelt 1994].

Since TSP is an NP-hard problem and broadly applicable, methods for obtaining a reasonably good solution (not necessarily optimal) are a continuous focus of studies. In this scenario, metaheuristic algorithms deserve to be highlighted because they can be adapted to different kinds of problems and usually lead to good quality solutions, provided that their hyperparameters are correctly fit. In this paper, we describe our work with a local-search-based metaheuristics, Iterated Local Search (ILS) [Helena R. Lourenço and Stutzel 2001].

Table 1. Percentage accuracy of TSP instances for Sequential and Parallel.

Instance	Sequential	Parallel
d198	96.91	91.97
a280	97.58	94.50
lin318	95.91	88.25
pcb442	98.18	94.95
rat783	94.98	91.17
pcb1173	93.79	87.55
fl1577	95.00	75.02

2. Iterated Local Search

This section describes the implementation of a sequential and a parallel version of the Iterated Local Search (ILS) algorithm with C++. The ILS relies on an embedded local search strategy, which iteratively provides locally optimal solutions within a given neighborhood structure. Inside the main loop, the local search is repeatedly applied with slight modifications to improve the solution, allowing exploration of different neighborhood spaces.

The solution assembling method focuses on improving solutions using the adopted local search strategy at each iteration, and this process can be highly time-consuming, particularly when numerous iterations must be executed. Therefore, aiming to reduce the total runtime of our algorithm, we proposed a parallel approach to balance solution quality with feasible running times. Our parallel ILS multi-core approach was based on the following idea: each CPU thread executes $\frac{N_{\text{iterations}}}{N_{\text{Threads}}}$ iterations of the original sequential loop. The proposed parallelization strategy allows each step to run simultaneously, and the solution is a minimum-cost tour between all the threads tours selected in a critical section, avoiding race condition.

3. Results

The code was compiled and ran on a machine x86_64, AMD Ryzen 5 5600GT with Radeon Graphics, 6 cores, and 12 threads, with Debian 12.2.0-14. The results are computed over an average of 30 trials for each instance with 1000 iterations of the main loop. The parallel code uses OpenMP with 12 threads. Our results were compared with the best available solution, maintained by [TspLib] and are presented in Tables (1) and (2).

As we can notice from the results, sequential and parallel ILS had an average accuracy of 96.05% and 89.05%, respectively, while the speedups were near 7.4. These values indicate that, although we had a slight loss of solution quality, we could accelerate the execution time, enabling the use of our proposed approach for larger instances, which are usual in real-world problems. The parallel solution achieved an efficiency of approximately 60%, indicating that while there is an improvement in computation speed, there is still space for optimization utilizing the available cores.

4. Conclusion

The ILS algorithm is conceptually simple and effectively solves the TSP. Using OpenMP, we reduced the execution time, compromising only a small percentage of the solution quality.

Table 2. Running time, speedup, and efficiency of each symmetric TSP benchmark for Sequential and Parallel ILS.

Instance	Seq. Time (s)	Par. Time (s)	Speedup	Efficiency (%)
d198	61.14	8.649	7.069	58.91
a280	175.5	24.11	7.279	60.66
lin318	277.5	39.41	7.041	58.68
pcb442	746.3	101.8	7.331	61.09
rat783	4620	599.9	7.701	64.18
pcb1173	16 760	2183	7.679	63.99
fl1577	42 930	5797	7.406	61.71

In future work, we will explore hybridization with other algorithms and optimizations tailored to specific hardware architectures, like GPGPUs and FPGAs, and use these methods in specific real-world applications. Additional improvements may include applying other methods for constructing initial solutions or employing machine learning techniques to guide local searches.

Agradecimentos: Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio ao desenvolvimento deste trabalho.

References

Helena R. Lourenço, O. M. and Stutzel, T. (2001). A beginner's introduction to iterated local search. *4th Metaheuristics International Conference*.

Reinelt, G. (1994). *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag Berlin Heidelberg, 1st edition.

Thomas H. Cormen, Charles E. Leiserson, R. L. R. and Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, 3rd edition.

TspLib. Available at: `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`. Acessed on: 11/11/2023.