Using Parallel Clonal Selection Algorithm to Solve Multi Travelling Salesperson Problem.”

by Dr.ayi Purbasari. St.,mt. Turnitin Paper -publikasi 12

Submission date: 19-Oct-2021 12:15PM (UTC+0700)
Submission ID: 1677843235
File name: 12USIN_1.PDF (2.28M)
Word count: 3368
Character count: 17969

2016 6th International Workshop on Computer Science and Engineering (WCSE 2016)

Edited by
Prof. Hayato Ohwada

ISBN 978-981-11-0008-6
Proceedings of 2016 6th International Workshop on Computer Science and Engineering (WCSE 2016)
Tokyo, Japan, 17-19 June, 2016, pp. 1 -6
doi: 10.18178/wcse.2016.06.001

Using Parallel Clonal Selection Algorithm to Solve Multi Travelling Salesperson Problem

Ayi Purbasari [+]

Informatics Department, Pasundan University, Bandung Indonesia

Abstract. This research focuses on creating a parallel Clonal Selection Algorithm (CSA), a parallel metaheuristic algorithm that used to solve the multi-Travelling Salesperson Problem (mTSP). CSA is an algorithm based on population and used in computational intelligent, including optimisation problem. After we identify the parallel potential for CSA and we construct a parallel algorithm with MPJExpress, a Message Passing Interface (MPI) library in the Java programming language. This code is verified by experiments with several datasets. We found that best cost will be reduced in line with the number of generation performed. The number of salesperson will improve the best cost achieved. The best cost is obtained with the lowest number of salesperson (2). The number of PEs is not significant in reducing the best cost. The best cost is obtained with the 2 processing elements. The execution time for the parallel version is greater than the serial time. This is because the experiments use the same execution environment. We need to explore other parallel potentials in this algorithm. This paper also shows that the best cost is obtained with the lowest number of salesperson. Furthermore, it is suggested that this proposed algorithm be used for all versions of the Vehicle Routing Problem (VRP) in future research.

Keywords: clonal selection algorithm (CSA), parallel clonal selection algorithm, multi-travelling salesperson problem (mTSP), MPJExpres.

1. Introduction

Multiple travelling salesperson problem (mTSP) is a generalisation of the travelling salesperson problem (TSP), where more than one salesperson is allowed to be used in the solution [1]. By incorporating some additional side constraints, mTSP is also possible to extend the problem to a wide variety of vehicle routing problems (VRPs). For years, there has been a lot of research and literature for the TSP and the VRP, but the mTSP has not received the same amount of attention. As an extension of TSP, Tang, Liu, Rong and Yang [2] solved the mTSP problem by converting the mTSP model into a single travelling salesperson problem (TSP) model.

TSP, which is a combinatorial optimisation problem, has many proposed optimisation algorithms. The class of exact optimisation algorithms have a good performance in many problems but are not efficient for solving larger scale combinatorial and highly non-linear optimisation problems [3]. Several algorithms, usually inspired by natural phenomena, some metaheuristic search algorithms with a population-based framework have shown satisfactory capabilities to handle high dimension optimisation problems.

Some researchers have used genetic algorithms (GAs) to solve TSP dan mTSP. Carter and Ragsdale [4] used a new GA chromosome and related operators for the mTSP and compared the theoretical properties and computational performance of the proposed technique to previous work. Kiraly and Janis [5] stated many approaches applied to finding a solution, including heuristic approaches published by Russel, Potvin et al. and Hsu et al. Zhang et al. used GAs for the solution of mTSP too. The first result can be bound to Zhang et al. converts the m-TSP into a single TSP and applies a modified GA to solve the problem.

[+] Corresponding author. Tel.: +628122311547; fax: +62222019371.
E-mail address: pbasari@unpas.ac.id.

Other researchers have used a clonal selection algorithm (CSA) for solving mTSP [6] CSA is the popular algorithms from Artificial Immune System (AIS) that using population-based and selection inspired from Clonal Selection Theory [6]. As a heuristic optimisation algorithm, CSA promises to solve the TSP problem [6], [7], [8] then used to solve mTSP by [10].

Either GAs or CSA are member class of metaheuristics search algorithm that provides suboptimal solutions in a reasonable time [6], [7]. Although the use of metaheuristics allows for a significant reduction of the temporal complexity of the search process the latter remains time-consuming for industrial problems [8]. Therefore, parallelism is necessary to not only reduce the resolution time but also to improve the quality of the provided solutions [9].

In this paper, a parallel version from a clonal selection algorithm is proposed for solving mTSP. CSA requires special representation and interpretable encoding to ensure efficiency. The aim of this paper is to review how CSA can be parallelised and applied to solve the mTSP problem. This research uses interpretable representation based CSA and uses the best technique for cloning and hypermutation phase, according to [10].

The paper is organised as follows. Section II reviews the literature consist of mTSP, the clonal selection algorithm and the parallelising algorithm. Section III contains implementation of parallel clonal selection algorithms to solve mTSP. Section IV briefly discusses the result of the algorithm implementation. Section V concludes the paper.

2. Literarture Review

2.1. Multiple travelling salesperson problem

The mTSP problem is defined on a graph G = (V,A), where V is the set of n nodes (vertices) and A is the of arcs (edges). $C = C_{ij}$ be a cost (distance) matrix associated with A. The matrix C is symmetrical if $C_{ij} = C_{ji}$, $\forall (i,j) \in A$. Here $x_{ij} \in \{0,1\}$ is a binary variable used to represent that an arch is used on the tour and c_m represents the cost of the involvement of one salesperson in the solution. The mathematical formulation for the m-TSP problem could be found in [1]. Fig. 1 shows the example of mTSP for solving the Vehicle Routing Problem (VRP).

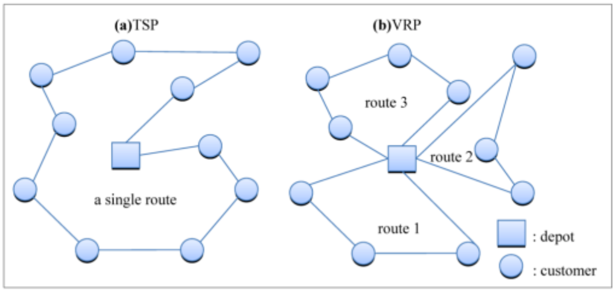


Fig. 1. Example of mTSP for VRP [11].

2.2. Clonal selection algorithm

In Artificial Intelligent (AI), Artificial Immune System (AIS) is a class of computationally intelligent systems inspired by the immune system, especially acquired immune response. There are four classes of AIS [12], including Clonal Selection Algorithm, an algorithm that inspired by the clonal selection theory (postulated by Burnet, 1957). The clonal selection theory inspired deCastro and Von Zuben to create clonal selection algorithms (CSA) named ClonalG [13]. CSA are used to solve many problems in computational intelligence, including optimisation problems like TSP[14], [15].

Bakhouya [16] and Chingtham [17] use CSA to find a solution for the TSP problem. Due successful to solve the TSP problem, Purbasari [10] uses this algorithm to solve mTSP. In particular, the research shows representation for tour and salesperson, and also shows several techniques for the cloning and hypermutation phase.

2.3 Parallel clonal selection algorithm

Like other population approaches, Clonal Selection Algorithm requires a significant amount of computation time. To reduce its computation time, many ideas have attempted to address this problem by adopting a parallel computation paradigm. Dabrowski and Kobale [18] use the parallel-CSA computation for the graph colouring problem. Hongbing *et al*. [19] apply the CSA parallelism for protein structure prediction using Open-MPI.

To build a parallel algorithm from nature inspired computation, we need two steps: 1) mapping between immune component and the problem, 2) building a parallel model using some principles and the concept of parallel computation design. Foster [20] shows some principles and concepts of parallel computation design. An immune mapping can be found in [21].

3. The Proposed Algorithm

This section will show the proposed algorithm for this paper. There are two sub-sections: clonal selection algorithm for mTSP and parallel CSA for mTSP. The first section describes the CSA for solving mTSP, whereas there is a hypermutation phase that is different from the hypermutation phase that used in algorithm for the TSP problem. The second section will show how to parallelise CSA and use it to mTSP.

3.1. Parallel clonal selection algorithms for MTSP

CSA starts from a group of initial population that represents a solution called the initial antibodies. Furthermore, affinity maturation is used to evaluate the performance of the solutions for each antibody. Each time the best solutions are chosen from the population according to the selection probability which is proportional to their affinity value. The best solutions are cloned and then hypermutated to produce new solutions of the next generation. These new solutions will replace the old solutions if they have better affinity. This procedure continues until it reaches a stop condition. Fig. 2 below shows the workflow of the Clonal Selection Algorithm technique.

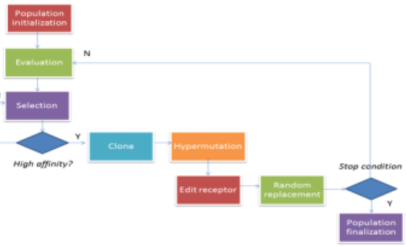


Fig. 2. Clonal selection algorithm.

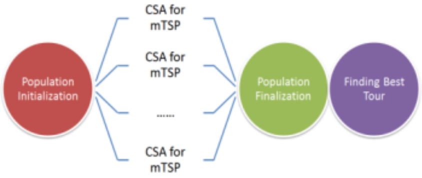


Fig. 3. Parallel CSA for mTSP

To solve the mTSP problem, we need several input parameters:

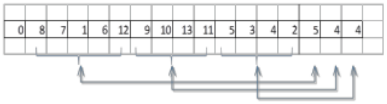
- Number of salesperson (s),
- Number of nodes/cities (n),
- Matrix that shows distances between cities (M),

Min and max number of cities for each salesperson. If we have 14 cities with 3 salesperson, then each salesperson will visit between 2 cities and $\lceil n/s \rceil$ cities. To parallelise mTSP, each salesperson will compute their best tours for 2 cities untul $\lceil n/s \rceil$ cities. Fig. 3 below shows parallel CSA for mTSP.

Fig. 4 below shows an example of a population that represents a solution. This population shows a tour with 14 nodes and node #0 as an initial tour.

0	8	7	1	6	12	9	10	13	11	5	3	4	2

Fig. 4. Example of a population that represents a solution.



0	8	7	1	6	12	9	10	13	11	5	3	4	2	5	4	4

Fig. 5. Example of two-part representation with 14 nodes and three salesperson.

We use this representation for the TSP solution, a tour with cost that optimal. To suit the needs of the mTSP problem, we enhance this representation by adding a part to represent salesperson. Fig. 5 above shows a representation of a tour with 14 nodes and 3 salesperson. This is called a **two-part representation**. The first part represents a tour and the second part represents salesperson. The first salesperson has his tour with 5 nodes, the second one has 4 nodes, and the third one has 4 nodes to visit. Fig. 5 shows an example of a two part representation. This two-part representation will affect the hypermutation process. There are some techniques for the hypermutation phase [10]. In this paper we will use the hypermutation phase by swapping nodes in a tour and also swapping salesperson's number of node from [10].

4. Implementation and Experiment Design

4.1. Implementation

This section describes the implementation of CSA for the mTSP problem. We used Java and the program is developed using Toshiba Portege Z935, Processor Intel® Core™ i5-3317* CPU@1.70GHz, memory 4GB, 64-bit Operating System and Windows 7. We used datasets from TSPLIB and picked burma14.tsp and vrp64.vrp dataset as examples. For parallel implementation, we used Java Message Passing Model Library, MPJExpress. As a developer, we could use the MPJExpress library to execute parallel Java applications on compute clusters or network of computers. MPJExpress is like a middleware for supporting a communication between the individual processors of clusters [22].

4.2. Experimenal design

Experiments were performed with the following steps:

- Dataset preparation: several datasets from TSPLib [23] e.g vrp64.vrp dataset;
- Defining parameters:

mTSP parameters:

- ○ number of nodes: 64;
- ○ number of salesperson: 2 - 34;

CSA parameters:

- ○ Population size (N) = 50;
- ○ The size of the selection (n) = 10;
- ○ The value of the cloning factor (β) = 0.1;
- ○ The value of the mutation factor (rho) 2.5;
- ○ Random size for replacement (d) = 5.

- Parallel implementation parameters: number of processor/ processing elements (pe) = 2, 4, 8;
- Determination of stopping criteria: number of generations (g) = 1000, 10.000, 100.000.

With the parameters that have been set, we determined the dataset and number of salesperson, which is from 2 until 34 salesperson. After the experiments have been performed, we will have mTSP solution, e.g best cost for each dataset with various numbers of salesperson, various numbers of nodes that a salesperson can visit, and various numbers of processing elements (pe). Besides that, we analyse parallel performances e.g. execution time.

5. Result and Discussion

This section provides the result of the experiment. We use vrp64.vrp dataset from TSPLib.

5.1. Result #1: best cost (serial and parallel version)

For best cost, we compare the number of salesperson = 2, 15, and 32. Fig. 8 below shows the best cost achieved for each generation. We can see from the graphs that the best cost will be reduced in line with the number of generations performed. The number of salesperson will improve the best cost achieved. The best cost will be obtained with the lowest number of salesperson (2).

For best cost, we compare the number of the processing element (PE) = 2, 4, 8 with the number of salesperson = 2. Fig. 9 above shows the best cost for each generation (for number of salesperson 2). We can see from the graphs below, this parallel version of clonal selection algorithm result same pattern with the serial version. Best cost will be reduced in line with the number of generations performed. The number of PEs is not significant to reduce best cost. The best is cost obtained with the lowest number of processing elements 2.

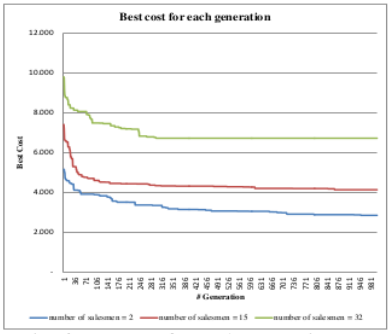


Fig. 6. Best cost for each generation.

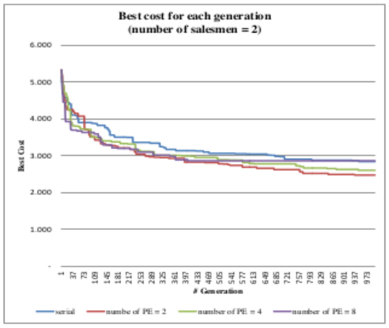


Fig. 7. Best cost for each generation (for number of salesperson 2).

5.2. Result #3: best cost for all salesperson with the number of generation – 100.000

This is the result of the best cost for all salesperson numbers, with 100.000 generation and a single version of CSA. Fig. 10 below shows the graph:

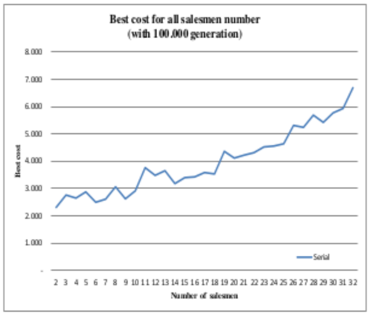


Fig. 8. Best cost for all salesperson number (with 100.000 generations).

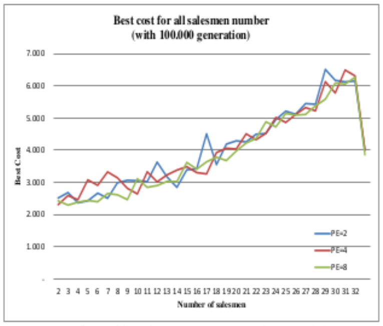


Fig. 9. Best cost for all salesperson number (with 100.000 generations and parallel versions).

With all number of salesperson, we see number of salesperson will improve best cost achieved. The best cost is obtained with the lowest number of salesperson (2). This is consistent with the graphs in Fig. 7 and Fig. 8. Using the same specification: 100.000 generation and all salesperson result for all number of processing elements (2, 4, 8); Fig. 9 shows the best cost achieved. Like the serial version of the algorithm, this parallel version shows that the number of salesperson will improve the best cost achieved. The best cost is obtained with the lowest number of salesperson (2). We can see from the graphs above that the best cost will be reduced in line with the number of generations performed. The number of PEs is not significant in reducing the best cost. The best cost is obtained with the lowest number of processing elements 8.

6. Conclusion

In this paper, a parallel version from clonal selection algorithm is created for solving mTSP. At first, we identify parallel potential for CSA and then construct a parallel algorithm. This code is verified by experiments with several datasets, for example vrp64.vrp. Through an experiment to find the best cost, we found that the best cost would be reduced in line with the number of generation performed. It is shown that a parallel version of CSA has the same pattern with the serial version.

The number of salesperson will improve the best cost achieved. The best cost is obtained with the lowest number of salesperson (2). The number of PEs is not significant in reducing the best cost. The best cost is obtained with the lowest number of processing elements 2. The execution time for the parallel version is greater than the serial time. This is because the experiments use the same execution environment. We need to explore other parallel potentials in this algorithm.

This paper also shows that the best cost is obtained with the lowest number of salesperson. Furthermore, using this proposed algorithm for all versions of the vehicle routing problem is suggested for future research.

7. Acknowledgments

I would like to thank the Department of Informatics and Technical Faculty of Universitas Pasundan (Unpas) Bandung Indonesia and the Directorate General of Higher Education Indonesia for all supports.

8. References

[1] T. Bektas, "The multiple traveling salesperson problem: an overview of formulations and solution procedures," OMEGA: The International Journal of Management Science, vol. 34, no. 3, pp. 209-219, 2006.

[2] Jonathan Timmis, "Artificial Immune Systems - Today and Tomorrow," Natural Computing, vol. 6, no. 1, pp. 1-18, 2006.

[3] Zahra Beheshti and Siti Mariyam Hj. Shamsuddin, "A Review of Population-based Meta-Heuristic Algorithms," *Int. J. Advance. Soft Comput. Appl*, vol. 5, no. 1, 2013.

[4] Arthur E. Carter and Cliff T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *European journal of operational research*, vol. 175, no. 1, pp. 246-257, 2006.

[5] András Király and Abonyi János, "A Novel Approach to Solve Multiple Traveling Salesperson Problem by Genetic Algorithm," *Computational Intelligence in Engineering*, pp. 141-151, 2010.

[6] Enrique Alba and Jose M. Troya, "A survey of parallel distributed genetic algorithms," *Complexity*, vol. 4, no. 4, pp. 31-52, 1999.

[7] Alsharhan S., J. R. Al-Enezi. Abbod M. F., "Artificial Immune Systems – Models , Algorithms and Applications," *International Journal of Research and Reviews in Applied Science (IJRRAS)*, pp. 118-131, May 2010.

[8] Enrique Alba and Gabriel Luque, "Evaluation of parallel metaheuristics," 2006.

[9] Enrique Alba, *Parallel Metaheuristic: A New Class of Algorithms*. A. John Wiley & Sons, Inc., 2005.

[10] Ayi Purbasari, Iping Supriana Suwandi, Oerip Slamet Santoso, and Rilla Mandala, "A New Approach to Solve Multiple Traveling Salesperson Problem by Clonal Selection Algorithm," *International Journal of Applied Engineering Research*, vol. 9, no. 21, 2014.

[11] Wan-Yu Liu, Chun-Cheng Lin, Ching-Ren Chiu, You-Song Tsao, and Qunwei Wang, "Minimizing the Carbon Footprint for the Time-Dependent Heterogeneous-Fleet Vehicle Routing Problem with Alternative Paths," *Sustainability*, vol. 6, no. 7, pp. 4658-4684, 2014.

[12] Jonathan Timmis and Emma Hart, "Application areas of AIS: The Past, The Present and the Future," in *4th International Conference on Artificial Immune Systems*, 2005.

[13] Leandro N. de Castro and Fernando J. Von Zuben, "Learning and Optimization Using the Clonal Selection Principle," *IEEE Transactions On Evolutionary Computation*, vol. 6, no. 3, pp. 239-251, June 2002.

[14] Jason Brownlee, "Clonal Selection Algorithms," Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information Communication Technology, Swinburne University of Technology, Melbourne, Australia , Technical 2009.

[15] Kulturel-Konak S. Ulutas BH, "A Review of Clonal Selection Algorithm and Its Applications," *Artificial Intelligence Review*, 2011.

[16] Gaber J. Bakhouya M., "An Immune Inspired-based Optimization Algorithm: Application to the Traveling Salesperson Problem," *AMO - Advanced Modeling and Optimization*, vol. 9, no. 1, pp. 105-116., 2007.

[17] Tejbanta Singh Chingtham, "Optimization of Path Finding Algorithm Using Clonal Selection: Application to Traveling Salesperson Problem," *International Journal of Computer Theory and Engineering*, vol. 2, no. 2, pp. 1793-8201, April 2010.

[18] Jacek Dabrowski and Marek Kubale, "Computer Experiments with a Parallel Clonal Selection Algorithm for the Graph," in *IEEE International Symposium on Parallel and Distributed Processing*, Miami, FL, 2008, pp. 1-6.

[19] Zhu Hongbing, Chen Sicheng and Wu Jianguo, "Paralleling Clonal Selection Algorithm with OpenMP," in *3rd International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, Shenyang, 2010, pp. 463 - 466.

[20] Ian Foster. (1995) Designing and Building Parallel Programs. [Online]. http://www.mcs.anl.gov/~itf/dbpp/

[21] Ayi Purbasari, Iping S. Suwandi, Oerip S. Santoso, and Rilla Mandala, "Data Partition and Communication On Parallel Heuristik Model Based on Clonal Selection Algorithm," *Telkomnika*, vol. 13, no. 1, pp. 193-201, 2015.

[22] Mark Baker, Bryan Carpenter and Aamir Shafi, "MPJ Express: towards thread safe Java HPC," in *IEEE International Conference on Cluster Computing*, 2006.

[23] TSPLIB. [Online]. http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/

ISBN: 978-981-11-0008-5

Using Parallel Clonal Selection Algorithm to Solve Multi Travelling Salesperson Problem.”

ORIGINALITY REPORT

19%	11%	18%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

6%

★ Ayi Purbasari. "Clonal Selection Algorithm parallelization with MPJExpress", 2016 8th Computer Science and Electronic Engineering (CEEC), 2016

Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		