An improved approach to resolve a combinatorial optimization problem based CoronaVirus Optimization Algorithm

Omayma El Majdoubi[1], Farah Abdoun[2] and Otman Abdoun[1]*

[1] Computer Sciences Department, FPL, Abdelmalek Essaâdi University, Larache, Morocco
[2] Research Center STIS, M2CS, ENSAM, Mohammed V University, Rabat, Morocco

Abstract. Combinatorial optimization problems refer to intractable problems that can't be performed using exact methods. The resolution of combinatorial problems geared towards the application of heuristics, metaheuristics also matheuristics, in order to provide good enough approximations. As exact methods provide resolution corresponding to small problem scale, the approximation methods target large scale of complex problems. Metaheuristics are used to deploy intelligent methods to solve complex problems in a reasonable amount of time. The performance of a metaheuristic is improved by means of parameters adjustment as well as, hybridization within heuristics, iterative improvement methods or various metaheuristics. The cooperation of several optimization algorithms leads to improve resolution, also to overcome the limitations reported in resolving NP-hard problems. The resolution of complex problems, is thus constrained by stagnation on local optimums, as the optimization process is possibly stagnant on a specific search space region. In fact, traveling salesman problem is a combinatorial problem, that arises problematics related to the efficiency of its resolution methods. The aim of this work is to investigate on the improvement of a new bio-inspired method so-called coronavirus optimization algorithm in order to provide improved resolutions to traveling salesman problem. Various intelligent approaches are investigated and hybridized within coronavirus optimization algorithm, namely random replicate operator, elitist selective replicate operator, iterated local search, stochastic hill-climbing also improved self-organizing map. The numerical results are obtained using symmetric TSPLIB benchmarks.

1 Introduction

The combinatorial optimization problems are NP-hard problems as they can be solved in a polynomial computing time. As, exact methods can handle only small scale problems, the approximation methods namely heuristics and metaheuristics, can achieve good enough solutions in a reasonable execution time. The efficiency of heuristics is dependent to each problem. Although, metaheuristics can deal with almost all the optimization problems. In fact, metaheuristic is an algorithmic structure deployed to solve optimization problems. Metaheuristics are approximate methods, that explore the search space using learning techniques to find satisfactory resolutions[1]. The tendance of exploration (diversification) and exploitation (intensification) are the main dilemmas of metaheuristics, that ensure search robustness[2]. The search exploration is based on memory preservation. Admittedly, metaheuristics are unable to solve all optimization problems. Hence, The hybridization of metaheuristics is an interference between algorithms components, by means of information exchange or operation on search processes[1]. The hybridization of different metaheuristics is described as a robust cooperative system that improves solutions efficiency. In addition, the adjustment of algorithm parameters has an important impact on occurring good solutions[3]. Importantly, the hybrid algorithms are advantageous by out-performing the limitations of individual algorithm. Thus, hybrid algorithms are better operating than each individual algorithm[4]. Generally, hybrid algorithms achieve better resolution through hybridizing mathematical methods or various metaheuristics[5].

2 Traveling Salesman Problem Formulation and Resolution

TSP is an intricate optimization problem, of which the objective is to find the shortest Hamiltonian path of all non-recurring existing visited cities and returned to the starting location. Various studies were carried out to solve TSP, still the problem is not solved completely[6]. The Euclidean traveling salesman problem (TSP) is defined as a complete weighted graph G=(V,E,d) where $V=\{1,2,\ldots,n\}$ is a set of vertices, $E=\{(i, j)x(i, j) \in V \times V\}$ is a set of edges and d is the weight value d_{ij} corresponding to the Euclidean distance linking the edge i to j. The objective function is to find a minimum of distance cycle in G wherein each vertex is visited once[7]. The applications of TSP are widely accurate on transportation, manufacturing of printed circuit board and planning[8,9].The TSP is a NP-hard problem of which the solution can't be performed using exact methods particularly when the number of cities is greater than 10 [10]. The theory of complete graphs assumes the existence of feasible solution related to TSP. Yet, proving the existence of a possible solution in

* Corresponding author: omayma.elmajdoubi@gmail.com

general graphs, is idem a NP-hard problem. In fact, the set of possible solutions is (n-1)!/2 [11]. Importantly, it is required to study large-scale TSP as it is a reference for other combinatorial optimization problems. The small scales of TSP are often resolved by means of exact methods namely, linear programming, dynamic programing and the branch and bound methods. Their application on large-scale complex problems leads to a combinatorial explosion and an expensive exponential resolution time. The approximate methods and heuristics are used to attain acceptable solutions and to avoid exponential computing complexity, by means of swarm intelligence, nearest-neighbor and greedy algorithms[12]. Tree-Seed Algorithm (TSA) is a new bio-inspired metaheuristic used to solve continuous combinatorial optimization problems, also developed to solve discrete symmetric TSP [13]. TSP is resolved by the use of iterated local search based on destroy-and-repair method, wherein iteratively the local search is linked to a perturbation method[14]. The evolutionary dynamics is hybridized within self-organizing maps, consisted of a mapping, evaluation and selection procedures to solve the Euclidean TSP. Particularly, the mapping operator is based on the nearest neighbors findings[15]. In addition, the bio-inspired metaheuristic is based on computed biological or social patterns . The artificial neural network is among the bio-inspired computing systems that simulates biological neural network for information processing[16]. The SOM training process is based on competitive learning responding to an input network pattern[17]. The SOM Algorithm is assigned for clustering of which the performance can be influenced by network topology and its connected units[18]. The SOM when applied to TSP, implies that SOM network is considered as a ring of neurons, admitting a set of cities as an input pattern. The SOM training process applies several iterations to the considered structure, throughout which a possible solution is determined. Then, the mapping operator is used and performed in order to find final admissible solution[7].

3 Improvements of Traveling Salesman Problem Resolution

The improvement of TSP resolution is applied by means of embedding an evolutionary algorithm within SOM algorithm. The considered memetic approach is consisted of constructive and improvement loops. Thus, the SOM process is used to perform local improvement in order to enlarge the set of neighbors interleaving in the constructive loop[15]. Genetic algorithms are efficient in solving TSP. Its efficiency is improved by means of applying a mutation operator based on stochastic hill climbing method[11]. The shortest path optimization is solved using iterative local search and hill climbing algorithms[19]. Stochastic hill climbing algorithm is based on local search approach that provides probabilistic perturbation, hence to avoid stagnant on local optima. The stochastic hill climbing algorithm admit a feasible solution, select an individual from its neighborhood, then evaluate their costs. If the condition of evaluation is satisfied, the current solution is replaced by its neighbor, otherwise, the initial solution is set to new candidate solution with a given probability[11]. Moreover, iterated local search is an iterative improvement heuristic, that provides a set of solutions[20], also performs on updating solution up till to reach a maximum number of evaluations or to be unable to find better resolution[19]. Generally, the iterated local search involves three main procedures namely: perturbation, improvement method and an acceptance criterion. The iterative local search provides candidate solutions by means of iterative disruptions and local improvements. The improvement method involves iterating improvements following some basic heuristics including simulated annealing. The disruption is used to introduce modifications and moves in candidate solutions, also to escape stagnant on a specific search space regions. The acceptance criterion occurs decisions to preserve candidate solutions during the search process. The main advantages linked to iterated local search are that candidate solutions consequent of disruption, induce better local resolution, also, the integration of iterative improvement methods leads to better exploration of the search space, compared to repetitive invoke of local search method starting from a random initial solution . Commonly, the local search involves the replacement of the current solution belonging to a specific search space by neighbors of the candidate solution. Neighbors are all the complete candidate solutions that can be obtained by applying improvement to a given solution[20]. Various local search heuristics are used to solve complex higher dimension problems[21]. The local search is based on fundamental methods for instance swap, insertion, deletion, replacement, perturbation[19]. Indeed, the permutation representation involves the application of local search methods and neighborhood search techniques in order to maintain the feasibility of the proposed solution. The neighborhood relations are as following : Transpose neighborhood is applied only if a permutation is obtained since the second by swapping two adjacent positions. Also, two permutations are 2-exchange neighbors as they are exchanging sequences since two random positions. The insertion neighborhood is also called shift neighborhood, is obtained by means of removing a permutation element from one position and inserting it at another position in the second permutation. The size of these processes are (n-1) corresponding to transpose neighborhood, also n.(n-1)/2 corresponding to exchange neighborhood and (n-1)!2 related to insertion neighborhood[22]. The improvement of local search strategy implies to explore the whole neighborhood and then applies accurate improvements[23]. For instance, variable neighborhood search is an iterative search strategy, that investigate improved distant neighbors by using local search to attain local optimum[24]. Furthermore, the TSP neighborhoods are commonly depending on the number m of vertices in the TSP instance. K-opt neighborhood is used to include all the different tours from the given one in at most k edges, reworded as applying m-opt move. The complexity of k-opt neighborhood is growing exponentially related to the growth of k [23].The iterative improvement methods arise the problematic of stagnation on a specific search regions. To remedy this limitations, it is appropriate to trigger occasionally the exploring of the neighborhood search space. The iterative improvement methods, through randomized decisions, provide search diversification and intensification[25]. The intensification of the algorithm is accomplished by means of local search operators, also diversification is carried out within heuristics and disruption methods[26].

4 CoronaVirus Optimization Algorithm : CVOA

The coronavirus optimization algorithm(CVOA) is a novel bio-inspired metaheuristic modelling the propagation of COVID-19. Patient-Zero is the first infected individual, that generates large populations of infected individuals, subjected either to reinfection, die or recover. The isolation measures, the die rate also the recovering are the main factors of gradual decrease of infected population. A parallel version of CVOA is developed wherein the evolvement is processed on each considered strain, in order to explore more search regions and minimize computing time. The CVOA methodology is described corresponding to single strain. Firstly, the initial solution is consisted of patient zero by randomization. Secondly, the disease spread is occurring deaths dependent to mortality rate, providing intensification by intermediate of new infected individuals that are not constrained to die and causing the infection spread according to a regular spreading rate or superspreading rate, also providing diversification by means of replicating new individuals according to spreading and traveling rates. The replicate functionality is dependent to the requirements of the optimization problem. Thirdly, updating population is required in each evolving iteration by deletion of deaths, appending previous infected individuals or isolated individuals to recovered population, joining recovered individuals to new infected population in the case of possible reinfection, also by gathering all the non-redundant infected population in new infected population. Commonly, the stop criterion is acquired if there is no more transmission of the virus or if a predefined number of iterations is attained[27]. The fig.1. is explaining the CVOA evolution:

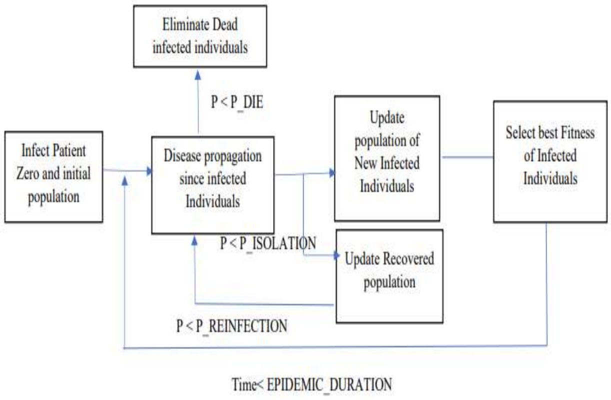

Fig.1. An illustration of CVOA evolution

5 Results and Discussions

This work consists on resolving the Euclidean TSP by implementing coronavirus optimization algorithm, also by applying iterative improvements in order to enhance fast convergence. The aim of this study is also an investigation to explore possible improvements of CVOA. The proposed implementation of CVOA is structured into sequential and parallel regions. The first sequential region consists on considering an initial random solution PZ by applying m-opt move such as m is the number of TSP nodes. The next region is a parallel block based on SPMD approach wherein the CVOA evolution is processed by each core of the machine, also the best solutions resulting of each iteration are preserved. The second sequential region handle the best solution consequent of each core execution. The processes manipulated on SPMD block, are training candidate solutions since random population based on applying m-opt move subsequently to initial PZ, also on generating new candidate solutions according to probabilistic functions as well as, on embedding new individuals since the replicate operator. The first experimentation applies random replicate operator in order to generate random set of individuals, since each given infected individual interleaving the main CVOA loop. The Table 1 is describing the results of the first experimentation. The results related to Table1, consequent of applying a random replicate operator to CVOA, leads to conclude that this method is efficient only for small or medium problem scale. The second experimentation is based on implementing an elitist selective replicate operator that starts by applying m-opt move on a given infected individual to generate new random population, also applies the fitness evaluation and elitist selection to select a portion of the best candidate solutions, in order to provoke disruptions in the CVOA evolvement.

The Table 1 describes the results related to the second experimentation. The application of an elitist selective replicate operator in CVOA, provides good approximations for small or medium data scale. In contrast the present conceived approach do not provide good resolutions. The third experimentation is an implementation of CVOA by intermediate of stochastic hill-climbing algorithm. Thus, the replicate operator generate initially random candidate solutions since a given infected individual, evaluate fitness of each corresponding individual. The cost of the new individuals are compared to the given infected individual. The infected individual is inserted in population within a specific probability , if its cost is less than the new individuals of the random population, else a new individual is inserted in the population obtained by means of applying neighborhood techniques on the infected individual. The Table 1 describes the results of the aforementioned approach. The present results prove that the stochastic hill-climbing algorithm is not efficiently improving the CVOA for large scale of TSP instances , although the large size of the search space. The present approach is based initially on random population, hence the results are also not improved. The fourth experimentation applies a hybridization of CVOA within iterated local search. The replicate operator is based on occurring candidate solution by applying m-opt move on a given infected individual, also by applying local search to the resulting individual. The local search evolves random population since the resulting individual as well as provides the best candidate solution since a specific search region. The Table 1 presents the results of the combination of CVOA within iterated local search strategy. The obtained results prove that iterated local search is not improving the TSP resolution, in particular for large scale of TSP instance. The CVOA is efficiently hybridized within SOM Algorithm, even though the improvement of the algorithm is required[28].The fifth experimentation of the present work deploys an hybridization of CVOA within Self-organizing maps using neighborhood search techniques. The replicate operator is used to train SOM Kohonen's network by applying m-opt move to PZ, also by applying neighborhood search techniques to the considered individual subjected to SOM training process. Also, the main neighborhood search techniques are applied in advance of main CVOA loop, in order to provide perturbations on candidate solution and to exempt local optimums and stagnant. The applied neighborhood relations are recombination, swap and permutation. The Table 1 is also describing the results corresponding to the proposed approach. The present numerical results prove an improvement of the TSP resolution compared to previous proposed approaches. The average deviation corresponding to Burma14, Ulysses16, Ulysses22, Rat99, Pr76 are 0.004, 0.014, 0.1, 0.58 and 0.36 respectively. The proposed approach is efficient in improving TSP resolution for small as well as large scale problems

Table 1. Numerical results of CVOA combined within random replicate operator(RRO), elitist selective replicate operator (ESRO), stochastic hill-climbing algorithm(SHC), iterated local search(ILS) and improved Self-organizing map (ISOM)

TSPLIB Instances	Present solution (RRO)	Present Solution (ESRO)	Present solution (SHC)	Present solution (ILS)	Present Solution (ISOM)	Exact Optimum
Burma14	32.8551	32.3282	31.3244	33.0757	31.0055	30.8785
Berlin52	22641	21358	21456	22112	19742	7542
Ulysses16	80.2092	79.0968	74.8964	80.6866	75.2144	74.1087
Ulysses22	104.9902	102.5563	85.6841	105.3634	83.2658	75.5975
Att48	109770	105900	106030	107660	104510	10628
Rat99	6500.1	6381.6	6460.7	6620.8	1913.9	1211
Pr76	466390	446380	441490	438210	147840	108159

Conclusion

The present investigation proves that hybridizing CVOA within improved learning strategy is reported efficient than heuristics and randomization. The efficiency is also related to the integration of neighborhood search techniques such as m-opt move, recombination, swap and permutation. The improvement of CVOA using SOM is also requiring high performances of parallel processing techniques.

References

1. Abdel-Basset, M., Abdel-Fatah, L., Sangaiah, A.K., *Metaheuristic algorithms: A comprehensive review.* In Intelligent Data-Centric Systems, Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications, 185-231(2018).
2. Blum, C., Roli, A., ACM Comput. Surv. **35**, 268-308 (2003).
3. Tezel, B.T., Mert, A., Expert Syst. Appl. **165**, 113976 (2021).
4. Lin, Y., Bian, Z., Liu, X., Appl. Soft Comput. J. **49**, 937-952 (2016).
5. Caserta M., Voß S., *Metaheuristics: Intelligent Problem Solving*. In: Maniezzo V., Stützle T., Voß S. (eds) Matheuristics. Annals of Information Systems, **10**. Springer, Boston, MA (2009).
6. Benyamin, A., Farhad, S.G., Saeid, B., Int. J. Intell. Syst. **36** ,1270-1303 (2021).
7. Wang, H., Zhang, N., Créput, J.-C., Neurocomputing **240**, 137-151 (2017).
8. Fujimura, K., Obu-Cann, K., Tokutaka, H., *Optimization of surface component mounting on the printed circuit board using SOM-TSP method*. In Proceedings of 6th International Conference on Neural Information Processing, ICONIP , 131-136 (1999).
9. Fujimura, K., Fujiwaki, S., Kwaw, O.C., Tokutaka, H., *Optimization of electronic chip-mounting machine using SOM-TSP method with 5 dimensional data*. In

Proceedings of International Conferences on Info-Tech and Info-Net: A Key to Better Life, ICII ,26-31 (2001).
10. Psychas, I.-D., Delimpasi, E., Marinakis, Y., Expert Syst. Appl. **42**, 8956-8970 (2015).
11. Katayama, K., Sakamoto, H., Narihisa, H., Math. Comput. Model. **31**, 197-203 (2000).
12. Wei, X., Ma, L., Zhang, H., Liu, Y., Alex. Eng. J. **60**, 189-197 (2021).
13. Cinar, A.C., Korkmaz, S., Kiran, M.S., Int. J. Eng. Sci. Technol. **23**, 879-890 (2020).
14. Archetti, C., Feillet, D., Mor, A., Speranza, M.G., Comput. Oper. Res. **98**, 24-37 (2018).
15. Créput, J.C., Koukam, A., Neurocomputing **72**, 1250-1264 (2009).
16. Pintea,C.-M., Intell. Syst. Ref. Libr. **57**, 1-186 (2014).
17. Jiang, Q., Wang, L., Hei, X., Cheng, J., Lu, X., Lin, Y., Yu, G., Neurocomputing **275**, 2716-2739 (2018).
18. Kuo, R.J., Rizki, M., Zulvia, F.E., Khasanah, A.U., Comput. Ind. Eng. **120**, 251-265 (2018).
19. Wijaya, J.S., Anggraeni, W., Muklason, A., Mahananto, F., Riksakomara, E., Djunaidy, A., Procedia Comput. Sci. **124**, 173-180 (2017).
20. Stützle, T., Ruiz, R. *Handbook of Heuristics* **1-2**, 579-605 (2018).
21. Zgurovsky, M.Z., Pavlov, A.A., *Studies in Systems, Decision and Control* **173**, 347-406 (2019).
22. Hoos, H.H., Stützle, T., *Stochastic Local Search.* In The Morgan Kaufmann Series in Artificial Intelligence, 417–465 (2005).
23. Karapetyan, D., Gutin, G., Eur. J. Oper. Res. **219**, 234-251 (2012).
24. Ramos-Figueroa, O., Quiroz-Castellanos, M., Mezura-Montes, E., Schütze, O., Swarm Evol. Comput. **53**, 100643 (2020).
25. Hoos, H.H., Tsang, E., *Foundations of Artificial Intelligence* **2**,135-167 (2006).
26. Mestria, M., Comput. Ind. Eng. **116**, 1-12 (2018).
27. Martínez-Álvarez, F., Asencio-Cortés, G., Torres, J.F., Gutiérrez-Avilés, D., Melgar-García, L., Pérez-Chacón, R., Rubio-Escudero, C., Riquelme, J.C., Troncoso, A., Big Data **8**, 308-322 (2020).
28. EL Majdoubi O., Abdoun F., Abdoun O., *A New Optimized Approach to Resolve a Combinatorial Problem: CoronaVirus Optimization Algorithm and Self-organizing Maps*. In: Motahhir S., Bossoufi B. (eds) Digital Technologies and Applications. Lecture Notes in Networks and Systems **211**, Springer, Cham, ICDTA ,947-957 (2021).