Meme-based Computational Optimization Framework

Felis Dwiyasa, Meng-Hiot Lim, Ren-Xiang Foo, and Shi-Wei Jason Teo

Nanyang Technological University, Singapore

Abstract From a computing perspective, a meme denotes information that represents knowledge, patterns, rules, or strategies used to solve complex problems. When applied on a problem, memes help a solver to arrive at good-quality solutions more efficiently, guiding the search process according to certain procedures or rules, instead of randomly searching through the solution space. Depending on the complexity of the problems, evaluating the suitability of memes and selecting a set of effective memes for different problems, however, are not straightforward tasks. A meme that works well for some problems may not be effective for other problems. Besides, different memes might have different degrees of importance in solving a problem. The level of importance of each meme might also change at different stages of the search. In this paper, we discuss how multiple memes can be generated and applied to solve computational optimization problems. A case study in combinatorial optimization is also presented and discussed.

1 Introduction

In the field of artificial intelligence, various learning paradigms have been introduced over decades. Regardless of the differences between various learning paradigms, learning always involves some information, knowledge, patterns, rules, or strategies that change the behavior of the learner.

The science and the art of selecting and incorporating memes within an evolutionary computing framework is the fundamental pillar of a memetic computational problem-solving paradigm (MCPP). Instead of searching for a solution from zero information, a MCPP capitalizes and exploits prior knowledge to guide the direction of the search. Each agent has access to a meme pool comprising various memes. Memes can also be passed on to another agent through propagation, selection, and variation mechanism (Ong, Lim, Zhu, & Wong, 2006) (Ong, Lim, & Chen, 2010) (Chen, Ong, Lim, & Tan, 2011). MCPP has been used to solve various problems, such as multi-objective scheduling (Yuan & Xu, 2015), vehicle routing problem (Meng & Pan, 2016), and inverse kinematics (Starke, Hendrich, & Zhang, 2018).

Despite the realization of memes as being the fundamental driver in complex problem-solving, memetic computing has often been simplistically alluded to as a form of hybridization. In its most basic and fundamental form, such hybridization is simply a combination of evolutionary algorithm and heuristic local search algorithm, otherwise referred to memetic algorithm or simply MA. Although hybrid evolutionary algorithm is a form of MA, its paradigm only fits the definition of the first generation of MA. The second generation of MA involves a pool of memes that go through selection and transmission mechanism. Through these mechanisms, memes that perform well will thrive whereas memes that do not perform well will diminish over time. In the third generation MA, the memes themselves can evolve into other types of memes.

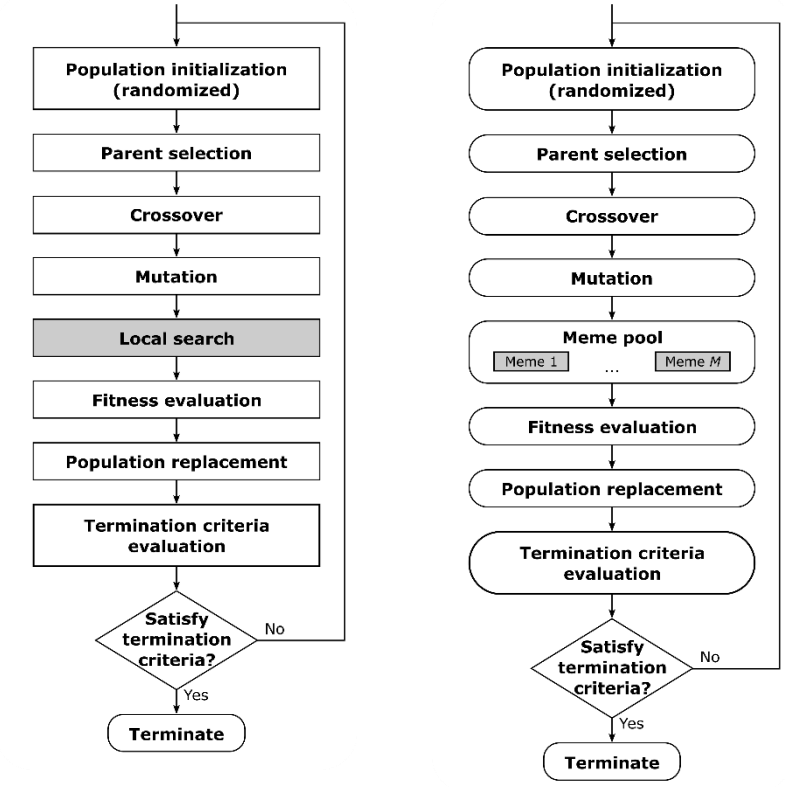
This paper is aimed at providing a computational optimization framework that is based on memes as being part of the evolvable and transferrable entities during the search. The memes are not limited to local search heuristics, but also other problem-specific information, knowledge, patterns, rules, or strategies. In addition, we propose how multiple memes can be incorporated within a MCPP.

The rest of this paper is organized as follows. Section 2 presents different ways of incorporating multiple memes into a search algorithm. In Section 3, we propose a systematic classification of memes that allows meta-learning framework to generate new memes from basic memes. Section 4 presents a case study of combinatorial optimization, which presents some commonly used basic memes and illustrates how new memes can be constructed. Finally, we present the conclusions and some future works in Section 5.

2 Multiple memes in evolutionary algorithms

Meme is a unit of information, knowledge, patterns, rules, or strategies incorporated within an agent. The first generation of MA is a simple hybrid of evolutionary algorithm and local search meme, whereas the next generations of MA involves evolution process on the memes (Meuth, Lim, Ong, & Wunsch, 2009).

Meme-based optimization algorithms may rely on either a single meme or multiple memes. Figure 1 (a) shows a simple hybrid algorithm where a single local search meme is applied at each generation. MA with a single meme is relatively easy to interpret and requires less computational resources. However, a single meme may not be enough to provide exploratory or exploitative orientation in the search process, resulting in ineffective convergence towards reasonable solutions. With multiple memes, the diverse range of the memes offers different paradigms on how to guide the search process, producing solutions with better quality as compared to a single meme. The term “multiple memes” in this paper is not to be confused with Multi-meme MA proposed by (Krasnogor, Blackburne, Burke, & Hirst, 2002). Here we use the term to describe a rich set of memes that can be used to guide MA as shown in the meme-based computational framework illustrated in Figure 1 (b).



(a) With a single local search meme (b) Incorporation of multiple memes

Figure 1 Computational framework - single meme vs multiple memes

Search algorithms with multiple memes are more complicated than the ones with single meme. There are also some design considerations on how the memes should be managed, as summarized in Figure 2.

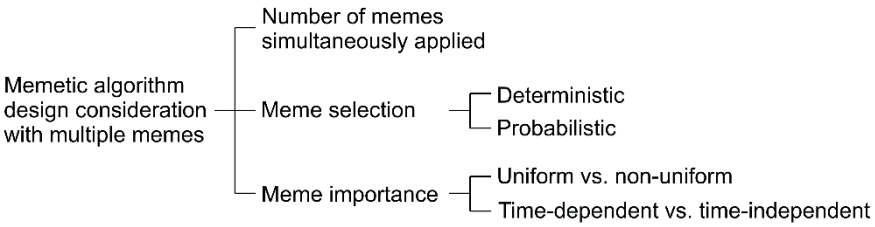


Figure 2 Design consideration of memetic algorithm with multiple memes

We may apply the memes one-by-one at each generation or altogether simultaneously. In between those two extremes, we may partially select some memes to be applied at each generation. Applying one meme at a time is straightforward and faster to compute, but the solution quality improvement is expected to occur at a slower rate. For M memes, if there is only one meme applied at each generation and each meme is given the same chance to be applied, the utilization rate of each meme would be $1/M$. Applying several memes simultaneously, on the other hand, increases the utilization rate of the memes. However, when applied simultaneously, the memes combined could have a positive or negative interaction with each other, and thus the higher utilization rate does not guarantee faster improvement of the solution quality.

Selecting a meme or a set of memes to be applied at each generation can be done using deterministic or probabilistic method. Some possible deterministic meme selection methods are random selection, round-robin procedure, weighted round-robin, and priority-based selection. As for probabilistic method, the memes are stochastically selected, where the probability of selecting each meme can be assigned *a priori* or obtained through meta-meme.

One of the most complicated issues involving multiple memes is deciding what memes are important and beneficial to be included. Although one may logically think of certain ways to solve a problem, it may turn out that it is not the best option available out there. As for ineffective memes, although the presence of them may not have a negative impact on the solution quality since inferior solutions are typically eliminated through selection biases, those memes would incur computational resources. Ideally, memes that are more important or effective should be applied more frequently, whereas the less important memes are applied less often.

The degree of importance might also change at different stages of the search. Some memes work more effectively at the beginning of the search or towards the end of the search, while some others are not affected by the stage of the search. Based on the characteristic of the degree of importance, we define the former type of memes as time-dependent and the latter ones as time-independent. The degrees of importance that are time-dependent would have a state variable that is determined by a certain time-dependent parameter.

3 Memes generation through meta-meme framework

The concept of meta-meme, which is “beliefs about beliefs” or “ideas about ideas”, can be traced back to the work presented by (Hales, 1997), which conceptualizes that an agent can have a certain perception about the perception of another agent and act according to that perception. This concept is intriguing since it models higher-order intelligence that resembles how living things perceive knowledge and create new knowledge. However, not many related works have been done to bring this concept closer into reality. (Meuth et al., 2009) proposes higher-order learning framework that contains a set of optimizer, memory, selector, and generalization mechanism to generate new memes. (Song, Lim, & Ong, 2011) uses neural networks to map problem features into control parameters to choose memes from a meme pool.

To date, MCPP implemented nowadays usually involves a static meme pool, where the set of memes in the meme pool are formalized before running the algorithm. Although the memes can be automatically selected (Krasnogor et al., 2002) and the parameters of the memes can be self-adjusted (Özcan, Drake, Altıntaş, & Asta, 2016), the memes in the meme pool themselves do not change.

Hypothetically, a true memetic approach should be endowed with the capacity to create new memes or having "memes that create memes". To some extent, this is similar to the concept of meme complexes (memeplex) introduced by (Chen & Ong, 2012), where the associations or synergies between memes are continuously updated and affect the probability of selecting a meme after another meme. The rationale behind the work is that consecutive activations of multiple memes might produce a unique effect that cannot be obtained when executing the memes individually. However, memeplex proposed by (Chen & Ong, 2012) solely puts emphasis on the consecutive activations of the existing memes rather than the generation of new memes.

We believe that the key to generate new memes lies in the ability to construct a complex meme from fragments of memes according to pre-established rules. This principle applies in language, for instance, where syllables construct words, words construct clauses, clauses construct sentences, and sentences construct paragraphs. The construction of the language requires categorization of words, also known as the parts of speech, where words have different roles in a sentence. Guided by grammar rules, words can be constructed into coherent sentences to communicate ideas.

Here we provide an example of classification of memes in computational optimization shown in Figure 3. This classification enables memetic algorithm to perform a mix-and-match procedure and create new variants of memes.

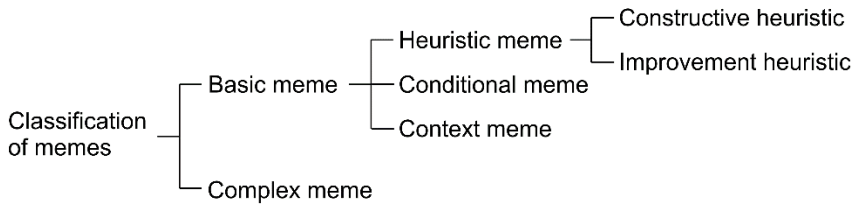


Figure 3 Classification of memes

Basic meme refers to the simplest unit of meme, whereas complex meme refers to a meme variant that involves two or more basic memes. This is analogous to the categorization of sentence structures, where a simple sentence contains one clause and a complex sentence contains two or more clauses. Basic meme can be categorized further into heuristic meme, context meme, and conditional meme.

Heuristic meme contains a procedure that defines how to build or improve a solution. There are two types of heuristic meme: 1) constructive heuristic, and 2) local search heuristic. Constructive heuristics start with an empty solution and builds the solution step by step until a complete solution is constructed. On the other hand, local search heuristics or improvement heuristics require a complete solution as a start and improves the solution by applying local changes according to a certain procedure.

Conditional meme is a meme that needs to be fulfilled before another meme can be applied. This type of meme cannot be applied on its own and must be attached to a heuristic or context meme. Suppose **X** is a conditional meme and **Y** is a heuristic or context meme, a complex meme that contains a conditional meme can be formulated as "If **X**, apply **Y**". With conditional memes, improvements can be specifically targeted on problematic areas rather than applied across the whole search space.

Context meme is a higher-level information that describes the expected characteristics of the problem or the solution. Some examples of context meme are acceptance strategies, search intensity, cluster information, and recurring patterns. Compared to heuristics, this type of meme holds higher-level information that assists the search process to produce relevant solutions.

We revisited the architecture of higher-order meta-learning proposed by (Meuth et al., 2009) where the higher-order meta-learning consists of several lower-order meta-learning blocks, recursively down to the first-order learning. With this architecture, we applied the principle of memes classification to allow new memes for computational optimization to be generated and evolved.

Suppose we have the first-order meta-learning components containing the basic memes of each type in the memory elements. The second-order meta-learning would construct complex memes from the basic memes that are optimized and generalized for the problem. In the third-order meta-learning, the complex memes produced by the second-order meta-learning can be further used to construct the more complex memes. The orders of learning can continue to increase up to *N*-th-order with the increase of the problem complexity and the required level of abstraction of memes.

4 Combinatorial optimization case study

Combinatorial optimization aims to find the best solution of combinatorial mapping from a finite set of objects. Some examples of combinatorial optimization problems are Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP), bin packing problem, and flow shop scheduling problem. Combinatorial optimization problems are encoded as integer permutation representation, where each gene in a chromosome in evolutionary algorithms contains a unique integer number.

Given a set of cities, the objective of TSP optimization is to find the sequence of cities to be travelled by a salesman such that the overall cost of visiting all the cities is minimized. The classical TSP specifies that each city must be visited only once, and the salesman starts and ends the trip at the same city. Figure 4 illustrates the typical GA chromosome representation of TSP, where each unique integer number represents a city. In this example, the salesman would start from city number 4, then go to city number 6, 2, 9, 7, 5, 1, 10, 8, 3, and head back to city number 4.

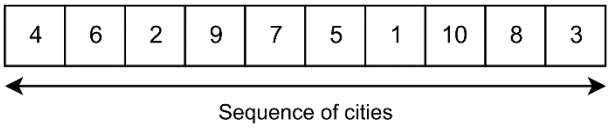


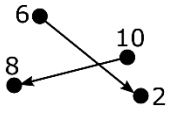
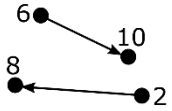
Figure 4 Chromosome representation of TSP

Both constructive heuristics and improvement heuristics can be used to solve TSP. We can use constructive heuristics to generate the initial chromosomes and apply improvement heuristics to improve the solution quality at each generation. The use of constructive heuristics for population initialization has been found to improve the solution quality tremendously as compared to random initialization (Paul et al., 2015).

As for improvement heuristics, one of the most commonly used heuristics for TSP is Lin–Kernighan (LK) or k-opt method proposed by (Lin & Kernighan, 1973), which is a generalization of 2-opt and 3-opt. The k-opt operator swaps some edges and selects the combination of edges that results in the shortest tour. According to performance tests conducted by (Helsgaun, 2006), the solution quality improves with higher *k*, but choosing *k* higher than 4 is not recommended due to its insignificant improvement in the solution quality at the cost of a higher computational complexity.

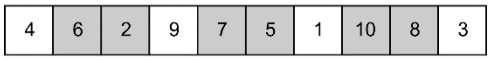
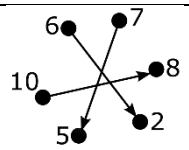
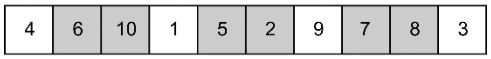
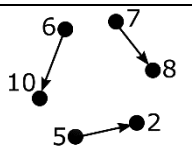
Table 1 illustrates the chromosome representation and tour visualization for 2-opt heuristics, where 2 edges are swapped to construct a new tour. Since there are only 2 edges to be swapped in 2-opt, there are only 1 possible alternative solution to be compared. If the alternative solution has a shorter tour length than the original solution, the alternative solution will be used in the next generations.

Table 1 Illustration of 2-opt heuristic for TSP

2-opt	Chromosome representation	Tour
Before	4 6 2 9 7 5 1 10 8 3	
After	4 6 10 1 5 7 9 2 8 3	

For 3-opt heuristic, there are 7 distinct combinations of tours after edge swapping (Ismkhan & Zamanifar, 2010). As a result, the computational complexity of 3-opt is higher than 2-opt, but in principle it can improve the solution quality faster. The chromosome representation and tour visualization of 3-opt is illustrated in Table 2.

Table 2 Illustration of 3-opt heuristic for TSP

3-opt	Chromosome representation	Tour
Before		
After		

The conventional way to apply k-opt heuristic is to systematically apply it across the whole search space or partially. Applying k-opt heuristic across the whole search space would undoubtedly produce a good solution, but the complexity increases exponentially with the problem size. Besides, after applying k-opt repeatedly, there will be a state when it is not able to improve the solution quality any further. If the heuristic is supplemented with a meme that could sense problematic areas that can be improved, the search process would be much more efficient.

To illustrate how the complex memes can be more efficient, consider a TSP tour illustrated in Figure 5 and three options of memes illustrated in Figure 6, where Meme A contains a basic meme and Meme B and Meme C contain complex memes.

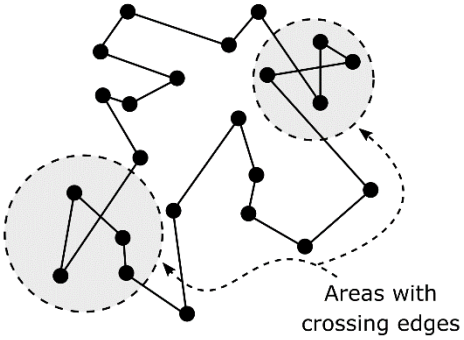


Figure 5 TSP with the presence of crossing edges

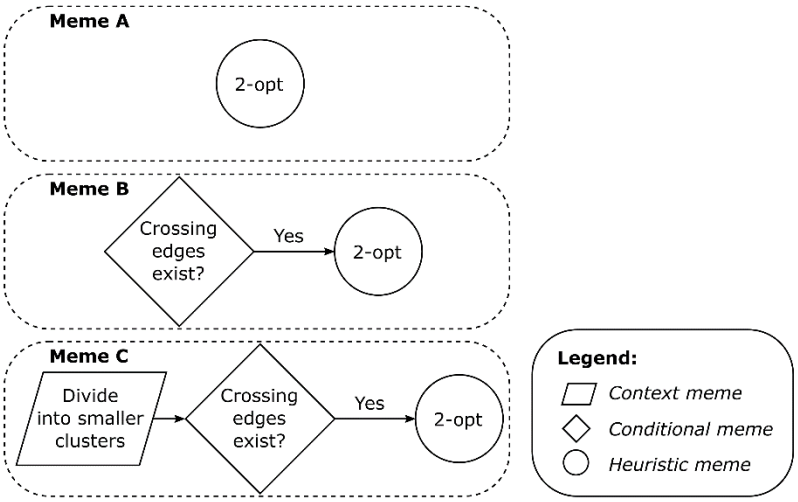


Figure 6 Alternative options of TSP memes

Multiple Traveling Salesmen Problem (MTSP) is a more generalized form of TSP, where the number of salesmen can be more than one. The typical objective function of MTSP is to balance the workload between salesmen or to minimize the total distance travelled (Carter & Ragsdale, 2006). MTSP is more complex than TSP since we need to determine both the assignment and the optimal sequence of the cities to be travelled by each salesman.

There are various chromosomal representation for MTSP, such as one-chromosome, two-chromosome, two-part chromosome techniques (Carter & Ragsdale, 2006), and multi-chromosome technique (Király & Abonyi, 2010). In this paper, we use a one-chromosome representation as shown in Figure 7.

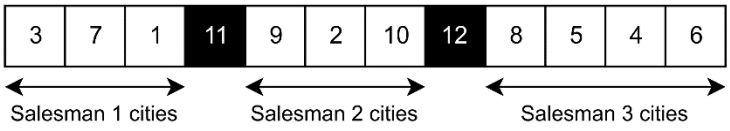
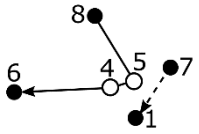
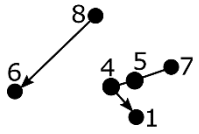


Figure 7 Chromosome representation of MTSP

Although TSP chromosome and MTSP one-chromosome representation look similar, applying k-opt heuristic is not effective to improve routes that belong to different salesmen in MTSP. A more effective way is to apply different heuristics for in-route and cross-route (Király & Abonyi, 2010), where in-route operator and cross-route operator are applied to improve the sequence of cities within one salesman and the routes of different salesmen, respectively. The cross-route operator used in the work is k-insert heuristic illustrated in Table 3, where a part of the route of one salesman is cut and inserted into the route of another salesman.

Table 3 Illustration of k-insert heuristic for MTSP

k-in-sert	Chromosome representation	Tour
Before	3 7 | 1 11 9 2 10 12 8 5 4 6	
After	3 7 5 4 1 11 9 2 10 12 8 6	

Rather than exhaustively applying k-insert heuristic, we can construct a complex meme, named as Meme E in Figure 8, containing k-insert heuristic and a context meme that finds the sections of the two routes that are nearby. Meme D illustrates the construction of a higher-order meme, where in-route condition is checked before applying Meme C for TSP in Figure 6.

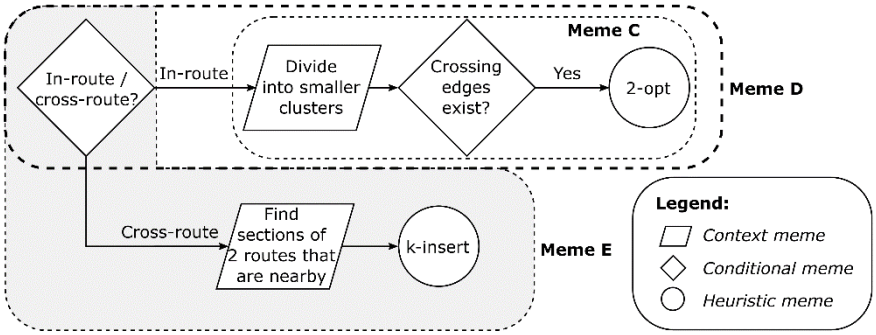


Figure 8 Example of MTSP memes

We may also generate some new variants of memes by simply changing one of the basic memes or the structure of the meme. Meme D, for instance, can be recreated into a new variant by changing the 2-opt into 3-opt, 4-opt, or other heuristic memes. The conditional meme that divides the problem into smaller clusters can also be differentiated by applying different clustering techniques.

5 Conclusions

A flexible framework for managing and incorporating memes into a search algorithm can uncover a whole new dimension of how computational optimization can be scaled up accordingly from the algorithm perspective. Rather than having the solutions generated through random evolutionary process or through a fixed set of memes, higher-level meta-learning architecture can be used to generate memes that might be more suitable to solve the problem at hand.

We illustrated the idea of memes generation in combinational optimization by showing how complex memes can be generated from basic memes. In principle, such memes are characteristically different and thus can be categorized. The compound memes of different categories can be further built into high-level complex memes. This demonstrates different levels of abstraction of memes and the possibility of creating new knowledge from some prior knowledge in memetic computation.

References

Carter, A. E., & Ragsdale, C. T. (2006). A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *European Journal of Operational Research, 175*(1), 246-257.

Chen, X., Ong, Y.-S., Lim, M.-H., & Tan, K. C. (2011). A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation, 15*(5), 591-607.

Chen, X., & Ong, Y. S. (2012). A conceptual modeling of meme complexes in stochastic search. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(5), 612-625.

Hales, D. (1997). Modelling meta-memes. In *Simulating Social Phenomena* (pp. 365-384): Springer.

Helsgaun, K. (2006). *An effective implementation of K-opt moves for the Lin-Kernighan TSP heuristic.* Roskilde University. Department of Computer Science,

Ismkhan, H., & Zamanifar, K. (2010). *Using ants as a genetic crossover operator in GLS to solve STSP.* Paper presented at the 2010 International Conference of Soft Computing and Pattern Recognition.

Király, A., & Abonyi, J. (2010). A novel approach to solve multiple traveling salesmen problem by genetic algorithm. In *Computational Intelligence in Engineering* (pp. 141-151): Springer.

Krasnogor, N., Blackburne, B., Burke, E. K., & Hirst, J. D. (2002). *Multimeme algorithms for protein structure prediction.* Paper presented at the International Conference on Parallel Problem Solving from Nature.

Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research, 21*(2), 498-516.

Meng, Z., & Pan, J.-S. (2016). Monkey king evolution: a new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization. *Knowledge-Based Systems, 97*, 144-157.

Meuth, R., Lim, M.-H., Ong, Y.-S., & Wunsch, D. C. (2009). A proposition on memes and meta-memes in computing for higher-order learning. *Memetic Computing, 1*(2), 85-100.

Ong, Y.-S., Lim, M.-H., Zhu, N., & Wong, K.-W. (2006). Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36*(1), 141-152.

Ong, Y.-S., Lim, M. H., & Chen, X. (2010). Memetic computation—Past, present & future. *IEEE Computational Intelligence Magazine, 5*(2), 24-31.

Özcan, E., Drake, J. H., Altıntaş, C., & Asta, S. (2016). A self-adaptive multimeme memetic algorithm co-evolving utility scores to control genetic operators and their parameter settings. *Applied Soft Computing, 49*, 81-93.

Paul, P. V., Moganarangan, N., Kumar, S. S., Raju, R., Vengattaraman, T., & Dhavachelvan, P. (2015). Performance analyses over population seeding techniques of the permutation-coded genetic algorithm: An empirical study based on traveling salesman problems. *Applied Soft Computing, 32*, 383-402.

Song, L. Q., Lim, M.-H., & Ong, Y.-S. (2011). *Neural meta-memes framework for managing search algorithms in combinatorial optimization.* Paper presented at the IEEE Workshop on Memetic Computing (MC).

Starke, S., Hendrich, N., & Zhang, J. (2018). Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation. *IEEE Transactions on evolutionary computation*.

Yuan, Y., & Xu, H. (2015). Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Transactions on Automation Science and Engineering, 12*(1), 336-353.