Vol.2015-MPS-105 No.13
2015/9/30

交叉手法を導入した並列コンサルタント誘導型探索アルゴリズム

飯田 修平[1] 中山 弘基[2] 長辻 亮太[1] 榎原 博之[1]

概要：コンサルタント誘導型探索 (CGS) は近年提案されたメタヒューリスティクスの一つである。この手法は仮想人間であるコンサルタントが解を構築するためのアドバイス提供して、同じく仮想人間であるクライアントがコンサルタントのアドバイスに基づいて問題の解を構築するアルゴリズムである。本研究では、CGS に遺伝的アルゴリズム (GA) における交叉と選択を導入した並列アルゴリズムを提案する。巡回セールスマン問題 (TSP) のベンチマークである TSPLIB の 6000 未満の都市数を持つ問題例に対して 3%未満の誤差率を達成することを示す。

キーワード：コンサルタント誘導型探索，交叉，並列処理，メタヒューリスティクス
組合せ最適化問題

Parallel Consultant Guided Search with Crossover

Abstract: Consultant Guided Search (CGS) proposed a recent metaheuristic methods. This approach is an algorithm in which a virtual person called a client creates a solution based on consultation with a virtual person called a consultant. In this study, we propose a parallel algorithm with a Genetic Algorithm's (GA) crossover and selection, and calculate an approximation solution for the Traveling Salesman Problem (TSP). We execute a computer experiment using the benchmark problems (TSPLIB). Our algorithm provides a solution with less than 3% error rate for problem instances using less than 6000 cities.

Keywords: Consultant Guided Search, Crossover, Parallel Processing, Metaheuristics, Combinatorial Optimization Problem

1. Introduction

A combinatorial optimization problem is used to determine a minimum- or maximum-valued combination based on the constraints given to evaluate the value of objective functions. Finding an optimal value(solution) is difficult without applying a combinatorial optimization problem. An approximate solution with a certain degree of accuracy is often acceptable instead, because in general, the solution can be obtained in a short period of time. It is well known that the execution time for determining an exact solution increases exponentially with the size of the problem. Therefore, obtaining an approximate solution sufficiently close to the optimal solution is faster than finding the exact value. Many methods for solving approximate solutions have been studied [1]. Metaheuristics, among the many approximate solution methods has been extensively studied in particular, because this method produces a general solution that can be adapted for many problems [2].

A Genetic Algorithm (GA) that mimics the proceses of natural selection is one of the most popular Metaheuristics [3]. On the other hand, Swarm Intelligence is a Metaheuristic that has been extensively studied as a method for solving optimization problems in recent years [4]. In addition, the Consultant Guided Search (CGS) algorithm has recently been proposed using a Swarm Intelligence algorithm [5][6][7][8]. This algorithm is inspired by the way real people make decisions based on advice received from

[1] 関西大学，大阪府吹田市山手町 3-3-25
[2] 株式会社日立システムズ

consultants. Human behavior is complex, but CGS uses virtual people that follow only simple rules. Also,in CGS, there is no leadership role in which to organize people; all people act on their own. Each virtual person is responsible for being both a client and a consultant. The consultant builds a strategy (solution) to lead the client to create a solution, and the client creates a solution based on the strategy that the consultant builds.

In this study, we find a better approximate solution for the Traveling Salesman Problem(TSP), by using a CGS method with a Crossover. The Crossover is a Genetic Algorithm(GA) programing method [3]. where the Crossover is a genetic operator, passing on the 'genes' of the parent to the child. Our proposed method in this study uses the Island Model solution [9] and the crossover for a consultant in CGS to build a strategy. In addition, the proposed method performs parallelization using MPI communication, on a PC cluster, in order to verify the effectiveness of our method.

2. Traveling Salesman Problem

The Traveling Salesman Problem(TSP) is the problem of finding the shortest possible distance in a cyclic route from a starting city, to each n city once, and back to the starting city.

When C_{ij} is the distance between city i and city j, and $V = \{1, ..., n\}$ is the set of n cities, the formula to minimize the objective function is as follows:

$$f(x) = \sum_{k=1}^{n-1} C_{x(k)x(k+1)} + C_{x(n)x(0)} \qquad (1)$$

$x(k) = i$ indicates that the kth city visited is i.

3. Consultant Guided Search

Consultant Guided Search(CGS) is a recent metaheuristic method which can directly exchange information between humans[5][6][7]. When a client decides an action, that action is sometimes based on advice from a consultant. CGS obtains the solution to a problem based on the relationship between the consultant and the client receiving the advice. The virtual person in this method plays the role of both the consultant and the client. The advice the consultant gives to the client is a solution, called a strategy. Since the virtual person acts as both the client and the consultant, the method is divided into modes to build a strategy as a consultant and to create a solution as a client. These modes are called sabbatical modes and normal modes, respectively.

Please refer to Iordache et al.[5] for more information about the parameters and search methods in the CGS. In our paper, we briefly describe the CGS of the algorithm as follows:

(1) The algorithm creates a virtual person, and sends that person to the sabbatical mode.
(2) In the sabbatical mode, each virtual person creates a solution according to a formula of strategy construction.
    In the normal mode, each virtual person creates a solution according to a formula of solution creation.
(3) The algorithm updates strategies after each virtual person generates a solution.
    If the solution is better than before in the sabbatical mode, it replaces a previous strategy.
    If the solution is better than the strategy the consultants used to create the solution in the normal mode, it replaces the previous strategy.
(4) The algorithm updates the strategies.
(5) If the consultant's reputation falls below a certain value, the consultant moves to the sabbatical mode.
    If the consultant builds a strategy a predetermined number of times, sabbatical mode changes to normal mode.
(6) If the algorithm meets the criteria, it terminates the search.

4. Related works

Iordache [10] proposed CGS-TSP to solve TSP. CGS-TSP combines CGS and a 3-opt method as a local search. In CGS-TSP-C, an improvement of CGS-TSP, the CGS consultant provides advice to the client adopting the concept of confidence. The evaluation experiment in this paper compared the proposed methods of CGS-TSP and CGS-TSP-C with the Ant Colony System (ACS) [11], and the Max-Min Ant System (MMAS) [12], and showed CGS-TSP and CGS-TSP-C taken local search effectiveness.

Ebara et al. [13] proposed a parallel hybrid algorithm that combines CGS and ACS. This study proposed a method in which the ant's pheromone information in ACS is taken over by the consultant's strategy as a virtual person in CGS and shared. This method included two-steps of ACS and CGS, the first step(phase 1) is the ACS search, and the second step(phase 2) is the CGS search.

In other CGS studies, Deepanandhini et al. [8] adapted the CGS to the Job-shop Scheduling Problem(JSP), and Iordache et al.[14] adapted CGS to the Quadratic Assignment Problem (QAP).

5. Proposed method

5.1 Summary

We propose an algorithm that performs in parallel with the island model and adapts the genetic algorithm's crossover and selection. The purpose of this paper is to create better accuracy is an approximate solution of TSP in a short period of time.

In general, a virtual person works as both the consultant and client. In this study, however, one virtual person works as only the consultant or only the client in CGS. A virtual person who works as consultant only builds a strategy and helps clients to a search solution. On the other hand, a virtual person who works as client only chooses a consultant and searches for solutions according to the consultant's strategy. Each virtual person has been allocated one processor core of a computer for parallelization. Our algorithm uses the island model for the parallel system.

In addition, the consultant's strategies create crossover and selection in the same island. Consultant strategy in CGS is independent from other consultant strategies. Accordingly, the proposed method periodically makes a pair of consultant strategies, and uses crossover and selection to generate new strategies.

5.2 Parallelization

In parallelization, one virtual person is allocated to one processor core of a computer. These virtual persons work as either consultants or clients and search for better solutions through the cooperation with other virtual persons who are either consultants or clients.

In our parallel method the number of virtual persons is lower than in other methods of CGS. For example, in Iordache's et al. method, the number of virtual persons is equal to $3 + 1400/n$ (n is the number of cities in TSP). In Ebara's et al. method, the number of virtual persons ranges from 20 to 30. In this study, because the number of virtual persons is the same number as the processor cores of the computers, one virtual person's computation ability is higher.

In general, because a virtual person of CGS works as both a consultant and client, the number of consultants and the number of clients is the same. In our method, because a virtual person works either as a consultant or client, the number of consultants and the number of clients can fit the problems variably. If the rate of consultants and clients is variable, we can adjust the balance of intensity and diversity.

One virtual person allocated to one processor core is regarded as one individual, and the population composed of the individuals is divided into subpopulations. In a subpopulation (one island), a consultant provides advice to a client and a client searches for a better solution based on this advice. Each island has a unique parameter. This parameter creates a solution for each island's identity. In particular, in the CGS parameter, the client chooses the consultant based on a parameter strategy or reputation, and the probability of using the consultant's advice in creating a solution is given a unique parameter in each island. The consultant periodically migrates to other islands taking it's own strategy and reputation. In this study, our method uses the ring-type island model with a pre-experiment. Fig.1 shows the island model of our algorithm.

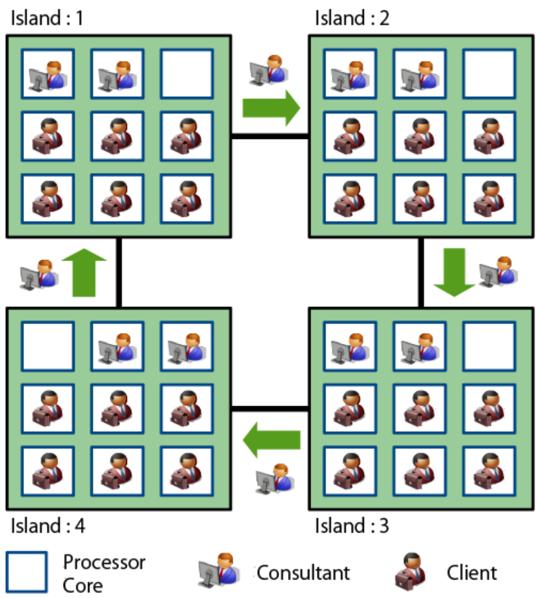

図 1 Island model of our algorithm

5.3 Crossover

Consultant strategy is independent of other consultants in CGS. Accordingly, for one consultant's strategy to affect another consultant's strategy, each consultant's strategy uses crossover and selection. Our proposed method uses the crossover Edge Exchange Crossover (EXX) method [15] proposed by Maekawa et al.. Because EXX is a crossover process using only a pair of the parents' edges, it is not the same cyclic route as made by the parents. Therefore, only the good part of parents' cyclic route are generated to the children. The following shows a specific example of the EXX method:

(1) A pair of two cyclic routes T^X, T^Y is sorted along the

order,

$$E^X = \{e^X_1, e^X_2, \cdots, e^X_n\}$$
$$E^Y = \{e^Y_1, e^Y_2, \cdots, e^Y_n\}.$$

The edge is $e = (e_S, e_T)$,

$$e^X_k = (e^X_{kS}, e^X_{kT}) = (t^X_k, t^X_{k+1})$$
$$e^Y_k = (e^Y_{kS}, e^Y_{kT}) = (t^Y_k, t^Y_{k+1}).$$

(2) Choose one edge $e^X_{i_1}$ from E^X, and choose $e^Y_{i_2}$ which has the same city as $e^X_{i_1}$.

(3) Choose edges $e^Y_{j_2S}$ such that $e^Y_{j_2S} = e^X_{i_1T}$, and choose $e^Y_{j_1S}$ such that $e^Y_{j_1S} = e^X_{i_2T}$.

(4) Exchange e^X_{j2S} and e^X_{j2S}. If $e^X_{i_1T} = e^Y_{i_2T}$, this operation ends.

(5) The part of cyclic route between edge $e^X_{i_1}$ and edge $e^X_{j_1}$ is

$$E^X_{i_1j_1} = \{e^X_{i_1+1}, e^X_{i_1+2}, \cdots, e^X_{j_1-1}\}.$$

Sort by the following cycle by reverse order.

$$\overline{E}^X_{i_1j_1} = \{\overline{e}^X_{j_1-1}, \overline{e}^X_{j_1-2}, \cdots, \overline{e}^X_{i_1+1}\}.$$

$\overline{e}^X_k$ is edge (e^X_{kT}, e^X_{kT}), the exchanging of cities.

(6) On part of the cycle route between e^Y_{i2} and e^Y_{j2}, make $\overline{E}^Y_{i_2j_2}$ and replace $E^Y i_2j_2$ with $\overline{E}^Y i_2j_2$.

(7) Set $i_1 = j_1, i_2 = j_2$ and then go back to (3).

In our algorithm, the consultant's strategy does a crossover with another consultant's strategy, and a new strategy is defined as the best strategy chosen from the parents and children. This algorithm is as follows:

(1) The consultant chooses another consultant in same island.

(2) The consultant sets its own strategy of cycle route T^X and choses the consultant strategy of the cycle route T^Y to the parents .

(3) The consultant applies EXX to T^X, T^Y. Set generated children to $T^{X'}, T^{Y'}$.

(4) The consultant chooses the best solution from $T^X, T^{X'}$ and $T^{Y'}$. At this time, don't choose from T^Y.

(5) If $T^{X'}$ or $T^{Y'}$ is chosen, the consultant replaces its own strategy.

This process runs in a period of time.

5.4 Algorithm flowchart

Fig.2 shows the proposed algorithm flowchart. The following description is about our algorithm:

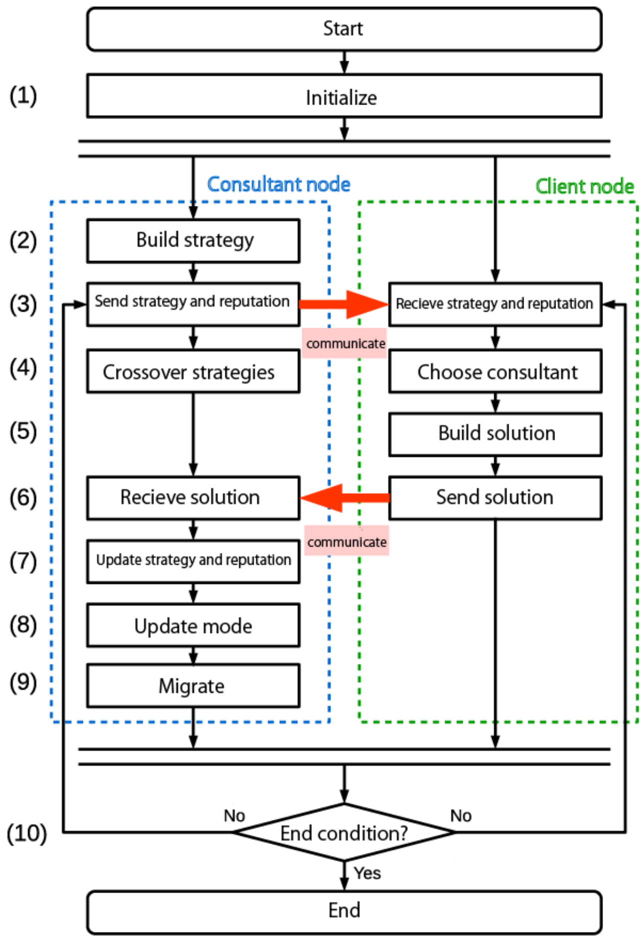

図 2 Proposed method's flowchart

(1) Initialize
All parameters initialize and virtual persons are defined.

(2) Build strategy
All consultant nodes build a strategy.

(3) Strategy and reputation communication
All consultant nodes send a strategy and reputation to client nodes in their islands. All client nodes receive a strategy and reputation from consultant nodes in their islands.

(4) Crossover and selection
All client nodes choose consultants who give the advice in the same island. In every certain number of iterations, all consultant nodes choose a consultant randomly in the same island and do crossover and selection.

(5) Build solution
All client nodes create solutions with the strategy of

consultant nodes.

(6) Send and receive solution
All client nodes send their own solution to the advising consultant node. The consultant node receives the solution from all clients who choose the same consultant.

(7) Update strategy and reputation
If a solution from a client node is better, the consultant set the solution as a new consultant strategy and update the consultant node's reputation.

(8) Update mode
If a consultant node's reputation is less than the threshold, the consultant node builds a strategy after deleting its own strategy.

(9) Migrate
Every certain number of iterations, and a fixed number of consultant nodes migrate to another island. The migrating consultant node is decided randomly.

(10) End condition
If the calculation time matches end condition, all nodes finish. When not matching the end condition, back to (3).

6. Experiment

6.1 Experimental method

TSP instances treated in this study are obtained from TSPLIB[16], distributing the TSP benchmark. This study creates an experiment for the following 8 problem instances, rat575, rat783, pr1002, u1060, u2152, pr2392, pcb3038, and rl5915. Table.1 shows each problem city size (problem scale), optimal solution, and our search time. We use a parallel system that is MPI environmental built by Score provided by the PC cluster Consortium[17]. Because this parallel system has one management server, and ten calculation computer nodes are connected by same LAN, each element can make MPI communication with each other. Table 2 shows the system's performance. Each calculation node has four processor cores. Therefore, this parallel system has a total of 40 processor cores.

Three evaluation experiments are performed to show our method's effectiveness. However, experimental results for the two methods of checking the island model effect and checking the crossover and selection effect are omitted.

This experiment compares our proposed method with the simple parallel CGS and ACCGS methods [13]. The simple parallel CGS is a parallel algorithm that runs a sequential CGS process independently.

表 1 City size and search limit time of problem instances

name	city size	optimization cost	search time[h]
rat575	575	6773	2
rat783	783	8806	2
pr1002	1002	259045	2
u1060	1060	224094	2
u2152	2152	64253	3
pr2392	2392	378032	3
pcb3038	3038	137694	5
rl5915	5915	565530	12

表 2 Performance of management server and calculation nodes

	management server	calculation nodes
CPU	Intel Xeon E5606	Intel Core i5-2400
	@2.13GHz x4	@3.10GHz x4
Memory	8GB	8GB
OS	CentOS 5.10	CentOS 5.10
MPI	SCore 7.0.1	SCore 7.0.1
number	1	10

表 3 Parameters of the proposed method

parameter name	value
$M_{consultant}$ (the number of consultants)	12
M_{client} (the number of clients)	28
β (distance weight)	5
a_0 (probability of choosing nearest city)	0.95
$fading\ ranks$	2
$Init\ reputation$	50
$Max\ reputation$	100
$Min\ reputation$	3
$Bonus$	5
$Migration\ number$	1
$Migration\ interval$	5000
$Crossover\ interval$	2500

表 4 Parameters of simple parallel CGS

parameter name	value
M (the number of virtual persons)	30
α (reputation weight)	7
β (distance weight)	12
a_0 (probability of choosing nearest city)	0.9
$fading\ ranks$	2
$Init\ reputation$	50
$Max\ reputation$	100
$Min\ reputation$	1
$Bonus$	10

Table 3 shows the proposed method parameters and island parameters. Table 4 shows the simple parallel CGS parameters. Each parameter is based on our preliminary experiment. The error rate is as follows:

$$Error\ rate = \frac{Solution - Optimal\ solution}{Optimal\ solution} \quad (2)$$

$Solution$ represents the cost of the solution as shown by the experiment, and the *Optimal solution* represents the cost of an optimal solution in TSPLIB.

6.2 Experiment to evaluate the proposed method

Table 5 shows the result of each problem instance solved by the proposed method and and the simple parallel CGS 10 times. In Table 5, for comparison, the result of the Ebara et al. method (ACCGS, [13]) is illustrated. The result of rl5915 by ACCGS was obtained from the experiment by Ebara's et al. method program [13] in the same environment. The average error rate and min error rate are calculated by eq(2) using the average cost and the minimum cost.

表 5 Experiment 1: comparison of results

name	algorithm	average	error rate[%]	minimum	error rate[%]
rat575	the proposed method	**6832**	**0.87**	**6810**	**0.55**
	the simple parallel CGS	6861	1.30	6850	1.14
	ACCGS[13]	6853	1.18	6828	0.82
rat783	the proposed method	**8916**	**1.25**	8898	1.05
	the simple parallel CGS	8994	2.14	8948	1.62
	ACCGS[13]	8920	1.30	**8882**	**0.87**
pr1002	the proposed method	262427	1.30	261917	1.11
	the simple parallel CGS	264797	2.22	264132	1.96
	ACCGS[13]	**261909**	**1.11**	**260981**	**0.75**
u1060	the proposed method	**226345**	**1.03**	**225742**	**0.74**
	the simple parallel CGS	229125	2.25	227798	1.65
	ACCGS[13]	228114	1.79	227333	1.45
u2152	the proposed method	65985	2.69	65824	2.45
	the simple parallel CGS	67446	4.97	66761	3.90
	ACCGS[13]	**65621**	**2.22**	**65380**	**1.88**
pr2392	the proposed method	**385184**	**1.89**	**384363**	**1.67**
	the simple parallel CGS	405769	7.34	402506	6.47
	ACCGS[13]	386072	2.13	384499	1.67
pcb3038	the proposed method	**141785**	**2.97**	**141004**	**2.40**
	the simple parallel CGS	157175	14.14	156339	13.54
	ACCGS[13]	156661	13.78	155358	12.83
rl5915	the proposed method	**578991**	**2.38**	**576595**	**1.95**
	the simple parallel CGS	618408	9.35	616764	9.06
	ACCGS	619172	9.49	614895	8.73

The proposed method obtains a better solution than the simple parallel CGS in all of the problem instances when comparing the proposed method to the simple parallel CGS. The difference in error rate is little in the small number of city problem instances, but the error rate difference is larger in the large number of city problem instances. This difference is considered that caused by the different parallel methods. Each processor core works a sequential CGS in the simple parallel CGS. For example, in Table 4, the number of virtual persons is 1200 (30 × 40), and the number of consultants and the number of clients is the same. In the proposed method, virtual persons work as 12 consultants and virtual persons work as 28 clients. In this way, the number of virtual persons in the proposed method is less than in the simple parallel CGS, therefore, it is possible to intensively search the solution space because one virtual person's computation ability is higher. Furthermore, because the virtual person's rate of the number of consultants is set less than the number of clients in the proposed method, we can place an emphasis on the client's search action. In the simple parallel CGS, because all virtual persons work as consultants and clients simultaneously, this method cannot make person adjustment.

In comparing the proposed method with ACCGS, our proposed method obtains better results than ACGS in the average error rate, except for pr1002 and u2152. When viewing rat575, u1060, pr2392, pcb3038 and rl5915, our proposed method obtains better results in the minimum error rate. Because ACCGS is an algorithm that combines CGS and ACS, this algorithm obtains better results than in the simple parallel CGS. However, ACCGS has, less improvement in pcb3038 and rl5915. On the other hand, because our proposed method is an algorithm combined with CGS, thus creating a new parallel method and crossover, our method obtains better results than in the simple parallel CGS in all problem instances. In particular, when a problem instance's city size is more than 3000 cities, there is a big improvement. In this case, the error rate shows more than a 10% improvement in pcb3038 and an error rate with more than 6% improvement in rl5915 between our proposed method and ACCGS. Thus, the proposed method handles large city size problem instances better. Furthermore, this result shows that CGS is a good match with the parallel method island model and the crossover of GA.

7. Conclusion

Our research proposed a parallel algorithm with a crossover to search for a better solution. The main features of our method are the allocation of a virtual person in CGS to a processor core; a parallel method of CGS in the island model, and a consultant strategy that interacts with other consultants by crossover. When our proposed method applied the TSP problem to instances under 6000 cities, the result shows that the proposed method derives a better solution than existing methods alone. Our proposed parallel method for CGS shows it can make search solution performance higher. Our proposed method obtained a better solution by adding crossover and selection. We believe that this addition increased the diversity of the solution.

謝辞 This work was partly supported by JSPS KAKENHI Grant Number 25330123, and the Information and Communication Technology Research Group for the Emergency Rescue Evacuation Support System of OR-

DIST Kansai University.

参考文献

[1] Sadiq M.Sait and Habib Youssef. *Iterative Computer Algorithms with Applications in Engineering*. IEEE Computer Society Press, 1999.
[2] Colin R. Reeves BSc, MPhil. *Modern Heuristic Techniques for Combinatorial Probrems*. Halsted Press, 1993.
[3] Talbi El-Ghazali. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.
[4] Ajith Abraham, Crina Grosan, Vitorino Ramos. *Swarm Intelligence in Data Mining*. Springer-Verlag New York Inc, 2006.
[5] Serban Iordache. Consultant-guided search - a new metaheuristic for combinatorial optimization problems. *In: GECCO 2010: Proceedings of the 12th Genetic and Evolutionary Computation Conference*, pp. 225–232, 2010.
[6] P.C. Pop and S. Iordache. A hybrid heuristic approach for solving the generalized traveling salesman problem. *LANZI, Pier L.(Hrsg.): GECCO, ACM*, pp. 481–488, 2011.
[7] S. Iordache. Consultant-guided search algorithms for the quadratic assignment problem. *Hybrid Metaheuristics*, pp. 148–159, 2010.
[8] D.Deepanandhini and T Amudha. Solving job shop scheduling problems with consultant guided search metaheuristics. *International Journal of Software and Web Sciences (IJSWS)*, Vol. 1, pp. 1–6, 2013.
[9] M. Ruciski, D. Izzo, and F. Biscani. On the impact of the migration topology on the island model. *Parallel Computing*, Vol. 36, No. 1011, pp. 555 – 571, 2010. Parallel Architectures and Bioinspired Algorithms.
[10] Serban Iordache. Consultant-guided search algorithms with local search for the traveling salesman problem. *Parallel Problem Solving from Nature, PPSN XI*, Vol. 6239, pp. 81–90, 2010.
[11] Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, pp. 53–66, 1997.
[12] Thomas Stützle and Holger H. Hoos. Max-min ant system. *Future Gener. Comput. Syst.*, Vol. 16, No. 9, pp. 889–914, June 2000.
[13] Hiroyuki Ebara, Yudai Hiranuma, and Koki Nakayama. Hybrid consultant-guided search for the traveling salesperson problem. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 97, No. 8, pp. 1728–1738, 2014.
[14] Serban Iordache. Consultant-guided search algorithms for the quadratic assignment problem. *Hybrid Metaheuristics*, Vol. 6373, pp. 148–159, 2010.
[15] Keiji MAEKAWA, Hisashi TAMAKI, Hajime KITA, and Yoshikazu NISHIKAWA. A method for the traveling salesman problem based on the genetic algorithm (in Japanese). *Transactions of the Society of Instrument and Control Engineers*, Vol. 31, No. 5, pp. 598–605, 1995.
[16] TSPLIB. http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/. (accessed 2015-8-31).
[17] PC Cluster Consortium. http://www.pccluster.org/. (accessed 2015-8-31).