International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

Solving the Quadratic Assignment Problem using the Swallow Swarm Optimization Problem

Safaa Bouzidi, Morad Bouzidi, Mohammed Essaid Riffi

***Abstract**: In recent years, there is a growing interest in swarm intelligent algorithms inspired by the observation of the natural behavior of swarm to define a computational method, which may resolve the hardest combinatorial optimization problems. The Quadratic Assignment Problem is one of the well-known combinatorial problems, which simulate with the assignment problem in several domains such as the industrial domain. This paper proposes an adaptation of a recent algorithm called the swallow swarm optimization to solve the Quadratic Assignment Problem; this algorithm is characterized by a hierarchy of search who allow it to search in a totality of research space. The obtained results in solving some benchmark instances from QAPLIB are compared with those obtained from other know metaheuristics in other to evaluate the performance of the proposed adaptation.*

***Keywords** : Combinatorial optimization problem, metaheuristic, Swarm Intelligent, Swallow Swarm Optimization, Quadratic Assignment problem.*

I. INTRODUCTION

The Combinatorial Optimization[1] is an emerging field of optimization in applied mathematics and computer science, it also related to Operational Research[2], algorithm theory, and computational complexity theory; Furthermore, it aims to use combinatorial techniques to resolve discrete optimization problems. A discrete optimization problem tries to find the best possible solution from a finite set of possibilities. Therefore, the Combinatorial Optimization has important applications in several areas such as artificial intelligence[3], machine learning[4], and software engineering[5].

The Quadratic Assignment Problem (QAP)[6] is a set of combinatorial optimization problems, it was introduced for the first time by Koopmans and Beckmann in 1957 as a mathematical model for location problems in economic activities[7], and in 1976, Sahni and Gonzalez proved that it belongs to the class of NP-hard problems[8]. This problem was applied in some real-life problems like hospital layout[9], Steinberg Wiring Problem[10] and scheduling parallel production lines[11].

In the literature, they are many metaheuristics adapted to resolve the Quadratic Assignment Problem. The metaheuristic is a generic strategy that define an algorithm, which find the optimum solution, within a reasonable amount of time, among these metaheuristics: the harmony search algorithm[12], the ant colony algorithm[13], Cuttlefish optimization algorithm[14], cat swarm optimization[15], the elephants herding optimization[16].

The rest of this paper is organized as follows: The first part will be devoted to the presentation of the Quadratic Assignment problem. The second part contains a description of methods. The third is an adaptation of Hybrid Swallow Swarm Optimization Algorithm to solve Quadratic Assignment problem. The fourth part presents a result and discussion, and finally a conclusion.

II. QUADRATIC ASSIGNMENT PROBLEM

The quadratic assignment problem can be described as a problem of assigning a set of facilities to a set of locations, with a given flow between every two facilities, and distance between every two locations in order to minimize the total assignment cost.

Mathematically, QAP can be formulated as given n facilities, n locations, and two $n \times n$ matrices, the flow matrix $F = f_{ij}$ and the distance matrix $D = d_{kl}$, where f_{ij} is the flow between facilities i for j all $i, j \in \{1,2,\ldots,n\}$ and d_{kl} is the distance between locations k and l for all $k, l \in \{1,2,\ldots,n\}$. We consider $\prod$ as the set of all possible permutations; the aim of this problem is to find the permutation $\pi_* = (\pi(1), \pi(2),\ldots,\pi(n)) \in \prod$ that minimizes the objective function:

$$\min_{\pi\in\Pi} = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} d_{\pi(i)\pi(j)} \quad (1)$$

III. SWALLOW SWARM OPTIMIZATION ALGORITHM

SSO is a bio-inspired algorithm[17], has been developed by Neshat.al[18] as a new swarm intelligence method[19] based on the behavior of swallow swarms. This algorithm is based on special characteristics of swallow that is revealed on some studies of the social behaviors taking into consideration swallows diversity, such as the very intelligent social life relation; the high-speed flying; and the hunting skills; and the migration of large groups. The colony of the swallow swarm is consisted of several sub-colonies, since that each swallow of the colony is responsible for something. They exist three kinds of particle:

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Safaa BOUZIDI*, Ph.D in Chouaib Doukkali **University**, Morocco.

Morad BOUZIDI, Researcher; Computer Science from Chouaib Doukkali University, Morocco.

Mohammed Essaid RIFFI, Professor, Computer Science, Chouaib Doukkali University, Morocco.

- The leaders particle (l_i), is divided in two types, Local Leader (LL_i), and Head Leader (HL_i). In each sub-colony when the explorer particle takes the best position in relation to neighboring particles, it will be selected as a Local Leader. Moreover, the best Local Leader is chosen as Head Local.
- The explorer particle (e_i), present the major population of the colony, and their primer responsibility is searching the appropriate places for resting, breeding, and feeding. This particle performs the search behavior under the influence of some parameters: the position of Local Leader and Head Leader, The best individual experience along the path, and the previous path.
- The aimless particle (o_i), which is the floating swallows that have the bad position of the search areas in the colony, since their fly randomly outside of the colony or between sub-colonies to search and inform the rest of the swarm.

The mathematical model consist that each particle is described by a position in colony and a velocity of flying, therefore their updated as follows:

$$V_{HL_{i+1}} = V_{HL_i} + \alpha_{HL} rand(\ )(e_{best} - e_i) + \beta_{HL} rand(\ )(HL_i - e_i) \quad (2)$$

$$V_{LL_{i+1}} = V_{LL_i} + \alpha_{LL} rand(\ )(e_{best} - e_i) + \beta_{LL} rand(\ )(LL_i - e_i) \quad (3)$$

$$V_{i+1} = V_{HL_{i+1}} + V_{LL_{i+1}} \quad (4)$$

$$e_{i+1} = e_i + V_{i+1} \quad (5)$$

$$o_{i+1} = o_i + \left[rand(\{-1,1\}) \times \frac{rand(\min_S, mas_S)}{1 + rand(\ )} \right] \quad (6)$$

Where:

V_{HL} = Velocity of lead leader,

V_{LL} = Velocity of leader local,

e_{best} = best position of the explorer particle,

$\alpha_{HL}, \beta_{HL}, \alpha_{LL}$ and β_{LL} are acceleration control coefficients adaptively defined[18].

e_i = current position of explorer particle,

V_{i+1} = Velocity of explorer particle,

O_i = current position of aimless particle.

IV. SOLVE QUADRATIC ASSIGNMENT PROBLEM

The development of DSSO algorithm to solve the QAP is based on adaptation of the operation and the operator to the QAP formula. For that, a redefinition operator method introduced by BOUZIDI [20] was applied to better capture the mechanism of the algorithm. The method presents a set of rules of define the data structure and the different operations between the data, which are defined as follows:

- The addition operator: The addition between the position p and velocity v is a new position p', which velocity translates the order of item the position. Thus, the result of addition between two velocities v and v' is a new velocity v'', which this action resumes to translation of v to v'.
- The subtraction operator: The subtraction between two positions is presented as a velocity.
- The multiplication operator: The multiplication between the velocity and coefficient is given by random swaps in velocity if random number between 0 and 1 is less than the coefficient, else it does nothing.

Moreover, The benefice is to change the search algorithms equations meaning from continuous to combinatorial space, and the application was proved a good result with particle swarm optimization for some combinatorial problems [20].

A solution of QAP is a set of permutation define the assignment for each factory to one location. The size of solution is n, where n is the number of location and factory, known also by problem dimension. The velocity is a difference between two solutions.

The pseudocode of general process of DSSO-QAP is summarized in Algorithm 1.

Firstly, the proposed adaptation DSSO generates randomly a set of population, as an initial solution. After that, the population is divided on a set of groups; one of them is dedicated for exploration (the last group G_d where d is number of groups) and others for exploitation $(G_i)_{1 \langle i \langle d}$.

Each solution s_i in G_d moved with random velocity generated for each instant (6), and compared with HL if it can be replace it, otherwise DSSP check if o_i can replace the worst LL of population.

For the exploitation solution which follows HL and there HL, they calculate the new position with equation (5), each solution compared the new solution if can be considered as HS or HL of this group.

Algorithm 1. Pseudo-code of DSSO-QAP

Begin
  Generate randomly a set of population P
  For each solution s_i in the P calculate the fitness
  Get HL of the population P
  Divided population in g groups P = (Gi), $i \in \{1...g\}$
  Get LL of each group g
  For each solution s_i in P do
    If (s_i in last group) then
      Applied random movement to s_i
      If($f(s_i) < f(HL)$) then
      HL replace the worst LL
      s_i will be the HL
      End if
    Else if(s_i is not a LL or HL) then
      Get LL of si
      Calculate V_{LL} of s_i
      Calculate V_{HL} of s_i
      Calculate Vi
      Move s_i according velocity Vi
    End if
  End for
End

V. NUMERIC RESULT AND DISCUSSION

The proposed adaptation has been implemented in C++ and compiled on a desktop PC, with Intel® Core ™ i5-4300M CPU @ 2.60 GHz speed and 4GB of RAM. As many as 58 instances of QAP problem have been extended from QAPLIB for test the efficient of the QAP-DSSO.

The instances on that DSSO has been tested including 13 Skorin-Kapov instances, all shown as the sample skoXXX; 22 uniformly generated Taillard instances, all shown as the sample taiXXX; and 15 all shown as the sample NugXXX and 8 benchemarks instances shown with several samples of RouXXX, ScrXXXand SteXXX. Also in all these models, X represents a size of instance, its ranger from 12 to100. Furthermore, the most selected instances are all real-life problems contributed by different authors to the QAPLIB. It is mentioned that the instances skoXXX and taiXXa are harder than the others. Moreover, for each instance, 30 independent runs of the algorithms are carried out.

The DSSO-QAP method consist of four parameters of acceleration control coefficient: $\alpha_{HL}, \beta_{HL}, \alpha_{LL}$ and β_{LL}. Based on literature the value of this parameters have been taken as 2.05 [21].

The table I summarize the experiments results obtained from running the algorithm on all of 58 benchmark instances of QAPLIB [22]. the first column shows the name of instance, the second column BKS indicate the best know solution of instance taken from the QAPLIB, the third and fourth column denote the best and the worst solution obtained by DSSO-QAP, and the next column show the average solution found. The SD column gives the stander deviation of all the result. Thus, the seventh and the eight columns show respectively the best deviation percentage from the Best Known Solution %DEVbest, and the value of %DEVavg represent the percentage measure of average deviation over 30 runs, this value is calculated by the formula:

$$\%DEVsolution = \left(\frac{Solutionlenght - BestKnowSolutionlenght}{BestKnowSolutionlenght} \right) \times 100 \quad (7)$$

Finally, the ninth column of table 1 reported the C1%/Copt where C1% indicates the number of solutions where relative error is less than 1 and Copt is the number of solutions equal to optimum known solution that means the number of iteration which its relative error is null. The last one of column denote the average time in seconds for the 30 runs.

Table- I: Numerical Results Obtained By DSSO Applied To Some QAP Instances of QAPLib

instance	BKS	Best	worst	Average	SD	%DEVbest	%DEVavg	C1%/Copt	TAvg(s)
Nug12	578	578	578	578	0.00	0.00	0.00	30/30	0
Nug14	1014	1014	1016	1015	0.99	0.00	0.10	30/16	0.03
Nug15	1150	1150	1150	1150	0.00	0.00	0.00	30/30	0.01
Nug16a	1610	1610	1622	1614.4	5.78	0.00	0.27	30/19	0.03
Nug16b	1240	1240	1240	1240	0.00	0.00	0.00	30/30	0
Nug17	1732	1732	1734	1732.4	0.80	0.00	0.02	30/24	0.03
Nug18	1930	1930	1938	1932.6	3.23	0.00	0.13	30/18	0.04
Nug20	2570	2570	2570	2570	0.00	0.00	0.00	30/30	0.03
Nug21	2438	2438	2442	2438.13	0.72	0.00	0.01	30/29	0.10
Nug22	3596	3596	3596	3596	0.00	0.00	0.00	30/30	0.06
Nug24	3488	3488	3506	3488.6	3.23	0.00	0.02	30/29	0.07
Nug25	3744	3744	3750	3744.33	1.16	0.00	0.01	30/27	0.14
Nug27	5234	5234	5234	5234	0.00	0.00	0.00	30/30	0.08
Nug28	5166	5166	5186	5170.4	7.33	0.00	0.09	30/22	0.21
Nug30	6124	6124	6148	6127.87	5.70	0.00	0.06	30/11	0.41
Rou12	235528	235528	235528	235528	0.00	0.00	0.00	30/30	0.01
Rou15	354210	354210	354210	354210	0.00	0.00	0.00	30/30	0.01
Rou20	725522	725522	728512	726452	854.79	0.00	0.13	30/4	0.52
Scr12	31410	31410	31410	31410	0.00	0.00	0.00	30/30	0
Scr15	51140	51140	51140	51140	0.00	0.00	0.00	30/30	0
Scr20	110030	110030	110748	110054	128.89	0.00	0.02	30/29	0.09
Sko42	15812	15812	15844	15815.3	8.01	0.00	0.02	30/22	2.80
Sko49	23386	23386	23538	23406.7	11.72	0.00	0.09	30/3	5.50
Sko56	34458	34458	34528	34476.7	16.78	0.00	0.05	30/2	7.18
Sko64	48498	48498	48620	48520.7	36.98	0.00	0.05	30/16	11.52
Sko72	66256	66256	66426	66313.2	40.45	0.00	0.09	30/4	32.16

Sko81	90998	91008	91164	91070.7	33.81	0.01	0.08	30/0	41.37
Sko90	115534	115542	115766	115659	70.48	0.01	0.11	30/0	62.41
Sko100a	152002	152002	152212	152097	53.42	0.00	0.06	30/1	138.58
Sko100b	153890	153890	154102	153961	51.96	0.00	0.05	30/3	149.79
Sko100c	147862	147862	148022	147906	41.17	0.00	0.03	30/3	110.82
Sko100d	149576	149578	149772	149688	57.13	0.00	0.07	30/0	137.02
Sko100e	149150	149150	149628	149198	93.27	0.00	0.03	30/13	71.26
Sko100f	149036	149036	149448	149172	76.40	0.00	0.09	30/1	137.07
Ste36a	9526	9526	9560	9536.53	10.54	0.00	0.11	30/12	1.21
Ste36b	15852	15852	15944	15855.1	16.51	0.00	0.02	30/29	0.69
Tai12a	224416	224416	224416	224416	0.00	0.00	0.00	30/30	0
Tai12b	39464925	39464925	39464925	39464925	0.00	0.00	0.00	30/30	0
Tai15a	388214	388214	390352	388390	448.81	0.00	0.05	30/24	0.05
Tai15b	51765268	51765268	51765268	51765268	0.00	0.00	0.00	30/30	0
Tai17a	491812	491812	496598	493598	1405.88	0.00	0.36	30/9	0.06
Tai20a	703482	703482	711760	707872	1920.47	0.00	0.62	26/1	0.13
Tai20b	122455319	122455319	122455319	122455319	0.00	0.00	0.00	30/30	0.04
Tai25a	1167256	1167256	1185890	1178370,27	4809.49	0.00	0.95	14/1	0.48
Tai25b	344355646	344355646	344355646	344355646	0.00	0.00	0.00	30/30	0.09
Tai30a	1818146	1825384	1842968	183439	4623.11	0.40	0.89	19/0	1.29
Tai30b	637117113	637117113	637117113	637117113	0.00	0.00	0.00	30/30	0.27
Tai35a	2422002	2435966	2458848	2447375	5724.28	0.58	1.05	14/0	4.36
Tai35b	283315445	283315445	283844480	283368349	158711	0.00	0.02	30/27	0.43
Tai40a	3139370	3165320	3189936	3176014,4	6330.7	0.83	1.17	7/0	9.67
Tai40b	637250948	637250948	637250948	637250948	0.00	0.00	0.00	30/30	0.56
Tai50a	4938796	4995292	5071770	5037603,27	19017.1	1.14	2.00	0/0	34.88
Tai50b	458821517	458821517	460630419	459093482	610992	0.00	0.06	30/25	2.80
Tai60b	608215054	608215054	608215054	608215054	0.00	0.00	0.00	30/30	4.48
Tai64c	1855928	185528	185528	185528	0.00	0.00	0.00	30/30	0.09
Tai80b	818415043	818415043	824737445	819283390	185464	0.00	0.11	30/19	38.53
Tai100a	21044752	21044752	21565794	21540521,3	17858.5	0.00	2.36	0/0	545.31
Tai100b	1185996137	1185996137	1186953819	1186215695	280461	0.00	0.02	30/14	73.65

Following the results obtaining in table 1, it can be noted that of all selected instances, the DSSO has been get nearly 88% of the BKS in a remarkable average time, and about 95% of the average deviation rate from BKS which are less than 0.5%. Therefore, the DSSO can indeed provide good solutions in reasonable time.

Moreover, owing to some similarity between DSSO approach and the Discrete bat algorithm(DBA) presented by Riffi et al.[22], the results of the comparison between the two approach has been given in Figure 1 : Comparison The average deviation between proposed algorithm and DBA, and figure 2 : Comparison The average time between proposed algorithm and DBA. These figures improve the DSSO is more efficient than the DBA in term of solution quality and the running time.

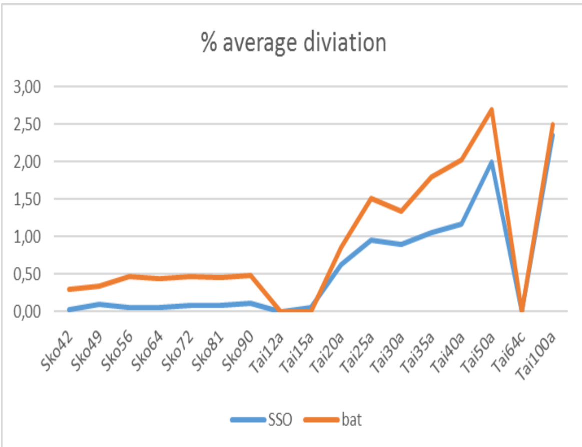

Fig. 1.Comparison The average deviation between proposed algorithm and DBA[22]

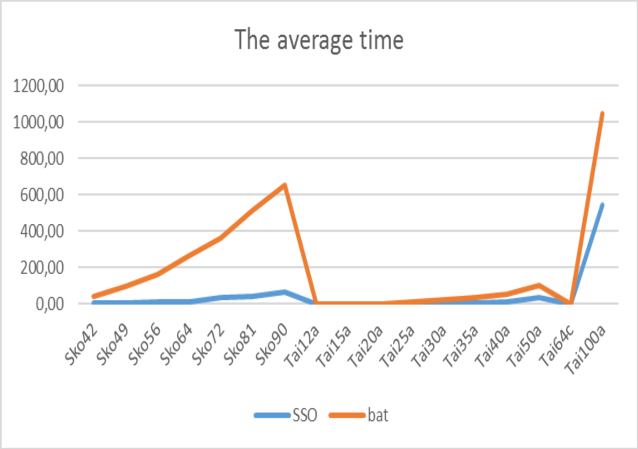

Fig. 2.Comparison The average time between proposed algorithm and DBA[22]

VI. CONCLUSION

The proposed work resume the resolution of the quadratic assignment problem with an adaptation of discrete swallow swarm optimization. The obtained result shows that the DSSO converge to optimal solution very speedy in remarkable times, this is due to the composition of different type of search in the population then particles are not stuck at the local minimum points easily. The comparison with other algorithms like bat algorithm gives more justification about the performance of adaptation that can be referenced to solve QAP. This result opens new aims to adapt the DSSO for other real problem.

REFERENCES

[1] K. Bernhard and J. Vygen, "Combinatorial optimization: Theory and algorithms," *Springer, Third Edition, 2005.*, 2008.
[2] R. L. Ackoff, "The Future of Operational Research is Past," *Journal of the Operational Research Society*, vol. 30, no. 2, pp. 93–104, Feb. 1979.
[3] S. J. Russell and P. Norvig, *Artificial Intelligence : A Modern Approach*. Malaysia; Pearson Education Limited, 2016.
[4] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, 2000, pp. 1–15.
[5] B. Barry, "Software engineering economics," *New York*, vol. 197, 1981.
[6] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment problems*. Springer, 2009.
[7] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica: journal of the Econometric Society*, pp. 53–76, 1957.
[8] S. Sahni and T. Gonzalez, "P-complete approximation problems," *Journal of the ACM (JACM)*, vol. 23, no. 3, pp. 555–565, 1976.
[9] P. M. Hahn and J. Krarup, "A hospital facility layout problem finally solved," *Journal of Intelligent Manufacturing*, vol. 12, no. 5–6, pp. 487–496, 2001.
[10] L. Steinberg, "The Backboard Wiring Problem: A Placement Algorithm," *SIAM Rev.*, vol. 3, no. 1, pp. 37–50, Jan. 1961.
[11] A. M. Geoffrion and G. W. Graves, "Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach," *Operations Research*, vol. 24, no. 4, pp. 595–610, 1976.
[12] M. Bouzidi and M. E. Riffi, "Adaptation of the Harmony Search Algorithm to solve the travelling salesman problem," *Journal of Theoretical and Applied Information Technology*, vol. 62, no. 1, Apr. 2014.
[13] N. Çetin Demirel and M. Toksari, *Optimization of the quadratic assignment problem using an ant colony algorithm*, vol. 183. 2006.
[14] M. E. Riffi and M. Bouzidi, "Discrete cuttlefish optimization algorithm to solve the travelling salesman problem," in *2015 Third World Conference on Complex Systems (WCCS)*, 2015, pp. 1–6.
[15] M. E. RIFFI and A. BOUZIDI, "Discrete cat swarm optimization for solving the quadratic assignment problem," *International Journal of Soft Computing and Software Engineering [JSCSE]*, vol. 4, no. 6, pp. 85–92, 2014.
[16] A. Hossam, A. Bouzidi, and M. E. Riffi, "Elephants Herding Optimization for Solving the Travelling Salesman Problem," in *International Conference on Advanced Intelligent Systems for Sustainable Development*, 2018, pp. 122–130.
[17] A. Darwish, "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 231–246, Dec. 2018.
[18] M. Neshat, G. Sepidnam, and M. Sargolzaei, "Swallow swarm optimization algorithm: a new method to optimization," *Neural Comput & Applic*, vol. 23, no. 2, pp. 429–454, Aug. 2013.
[19] *Swarm Intelligence: From Natural to Artificial Systems*. Oxford, New York: Oxford University Press, 1999.
[20] S. Bouzidi, M. E. Riffi, M. Bouzidi, and M. Moucouf, "The Discrete Swallow Swarm Optimization for Flow-Shop Scheduling Problem," in *International Conference on Advanced Intelligent Systems for Sustainable Development*, 2018, pp. 228–236.
[21] Y. Marinakis, A. Migdalas, and A. Sifaleras, "A hybrid Particle Swarm Optimization – Variable Neighborhood Search algorithm for Constrained Shortest Path problems," *European Journal of Operational Research*, vol. 261, no. 3, pp. 819–834, Sep. 2017.
[22] M. E. Riffi, Y. Saji, and M. Barkatou, "Incorporating a modified uniform crossover and 2-exchange neighborhood mechanism in a discrete bat algorithm to solve the quadratic assignment problem," *Egyptian Informatics Journal*, vol. 18, no. 3, pp. 221–232, 2017.

AUTHORS PROFILE

Safaa BOUZIDI is Ph.D in Chouaib Doukkali University, Her research interests emphasize the search and the application of swarm intelligence for combinatorial optimization problem.

Morad BOUZIDI is a researcher; he obtained his PhD on Computer Science from Chouaib Doukkali University in Morocco. he has published several papers in various peer reviewed journals/conferences of international repute. his research interests are the general metaheuristic algorithms based on swarm optimization nature inspired, he is particularly interest is understanding the impact of a different technique and function on a several optimization combinatorial problems.

Mohammed Essaid RIFFI is a professor in department of computer science at the Chouaib Doukkali University, and member of the Laboratory of Research in Optimization, Emerging System, Networks and Imaging (LAROSERI). He has published 44 papers in various reputable national/international journals and conferences. His research focus is study and implement of metaheuristic methods for a various optimization problems for continue and discrete domain.