Development of offshore maintenance service scheduling system with workers allocation

Guilherme Angelo Leite
Electrical Engineering Department
PUC-RIO
Rio de Janeiro, Brazil
guialeite92@gmail.com

Marley Maria Bernardes Rebuzzi Vellasco
Electrical Engineering Department
PUC-RIO
Rio de Janeiro, Brazil
marley@ele.puc-rio.br

***Abstract*—In order to develop an offshore maintenance schedule support system, this work presents a new model for constrained combinatorial problems: CPSO+. This model is a combination of two previous models: the PSO+, which presented good results in problems with nonlinear constrains; and the CPSO, which is an adaptation of PSO for application in combinatorial problems. The proposed model has been adapted to solve the complex problem of defining the best sequence of offshore maintenance activities and allocated staff to maximize service provider profitability within three months, while respecting all service completion time constraints and specific offshore work constraints. To evaluate the performance of this new model in solving the proposed problem, two CPSO+ variants were evaluated against the original CPSO model, in six proposed simulation cases. The results of the simulations indicate that the proposed CPSO+ model with reduced initialization variation outperforms other evaluated models in execution time and solution quality to the given problem.**

***Index Terms*—Particle Swarm Optimization, Optimization, Combinatorial Problem, Offshore Maintenance, Combinatorial PSO.**

I. Introduction

The oil and gas industry is an economic strategic sector in Brazil. Therefore, the solution to relevant problems in this field is highly important to the country. In this context, one of the challenges of this sector is to replace the manual and archaic procedures for modern, automated and data driven techniques, specially on the scheduling segment [1]. Among the different support areas associated to offshore oil extraction, the main department is the offshore maintenance team, which is responsible to ensure the equipment's correct setup and performance. In this segment, the most outdated process is the workers boarding scheduling, with no proper tool to assist on this task which considers the necessary restrictions.

This problem is known as Flexible Job-shop Scheduling Problem, an optimization problem to solve machine and resources allocation in tasks with known restrictions [2]. Today, on the Brazilian industry, some companies still operate without this kind of software support, depending, instead, on expert decisions. Others, utilize generic software that does not analyze the offshore work specific restrictions, resulting in sub-optimal solutions that request further manual tuning to be implemented on the field. Even though the companies operates in that way, this process can take several days of different specialists and is highly susceptible to human errors.

Since there are no techniques specifically designed for offshore restrictions, this study uses as reference other industry segment algorithms with similar restrictions. Those segments are: the problem to balance the data process on multiple cores [3]; the information path selection on network environments with bandwidth limitation [4] and papers on modeling Job Shop Scheduling Problems [5]–[7]. These studies support that the Flexible Job Shop Scheduling Problems, the literature name for the presented offshore problem, is a relevant problem to be studied, especially with multiple workers and constrains. The Particle Swarm Optimization (PSO) [8] algorithm has presented good results both in terms of processing speed and quality of the solution for high dimensional problems with constrains, so it has a good fit with the proposed problem. Although, the model adaptations does not cover all applications yet. Variations of the original PSO have been developed, such as PSO+ [9], which extends the PSO to deal with nonlinear restrictions using a multiple swarm logic, and the CPSO [10], which is suitable for combinatorial optimization problems. However, there is no PSO version that incorporates both capabilities.

Therefore, the main objective of this work is to propose a new Flexible Job Shop Scheduling Problem optimization model specifically designed for the offshore maintenance sector restrictions. This new model, called CPSO+, is inspired on the PSO+ high performance tools and the combinatorial adaptations from the CPSO model, allowing it to benefit from the advantages of both models.

II. Offshore Maintenance Scheduling

Oil platforms in Brazil are mainly offshore constructions similar to ships that should never leave the sea. Although similar to normal ships, the offshore platforms need a variety of services to keep operation, such as food supply, accommodation facilities, electricity, transportation to land, loading / offloading, telecommunications, medical services, and maintenance, safety and emergency equipments [11]. Even though those services are similar to onshore services, they are regulated by different laws dedicated to offshore jobs. Those regulations impose that:

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

- The maximum work hours per day is 12h, with 1h breaks for the meals;
- The maximum time on board is 15 days;
- the worker has 24h off for each day on board.

Adding to all the law restrictions, the offshore market has some other standards in the segment as pre-scheduled dates to access the platform, limit to on board headcount, specific safety train and periodic medical checkups [12]. Therefore, the focus of this study, the offshore maintenance, is a hilly constrained multidimensional problem that should be optimized over a key result.

The offshore maintenance scheduling team key result is to maximize the profit for the next three months respecting all the sector restrictions. Naturally, those profit is correlated to the number of completed services, so, to maximize the profit, the team focus in minimizing the estimated finish time from the client selected services. Therefore, the maintenance scheduling team is responsible for:

- choosing the order to execute each job;
- splitting the clients' jobs in tasks of maximum 12h;
- selecting the workers for those tasks;
- defining when each worker should board and disembark the platform.

All these tasks should be scheduled respecting all mentioned segment restrictions.

III. Flexible Job Shop Scheduling Problem

In the literature, there are different variants on the Job Scheduling Problem. The main blocks relevant to this study are: resource allocation on jobs, known as Flexible Job Scheduling Problem [13]–[15], and resource sharing, known as Parallel Job Scheduling Problem [16], [17].

A. Order decoder

The other transformation necessary to optimize scheduling problems is to decode the order vector. Given a task sequence, this deterministic function converts it on a feasible schedule, start and end date for each task, that respects the workers restrictions and the tasks precedence conditions.

In this study, those conditions are:

- An offshore task should start after a boarding day;
- Offshore tasks should be concluded before the 15th day on board of the selected worker;
- The break after landing is equal to the period on board;
- Each boarding period has a limit of worker on board;
- Onshore tasks should be executed when the worker is on land, but after the break period;
- Each worker only execute one task at a time.

Although most of the restrictions are handled by the order decoder, the desired deadline for the job still need to be evaluated after the schedule is constructed. Therefore, the job deadline is the only restriction that can evaluate a optimization result as invalid.

IV. Particle Swarm Otimization (PSO)

It's proven that the Flexible Job-shop Scheduling Problems are classified as NP-Hard problem, a problem that is not computational time solvable testing all possible solutions when it increases in dimension [18]. Those type of problems, then, requires techniques that explores the search space in a semi-random process using the acquired knowledge to redirect the algorithm towards the best solution, the meta-heuristic methodologies [19].

As the NP-hard problem significance is increasing in the industry, the meta-heuristic algorithms are also conquering more field on the literature. Even though those methods does not ensure mathematical convergence on a global optimum, there results are considered satisfactory for real case application due to their convergence strategy [20]. There are multiple techniques and variations developed to outperform the others in it's specific field, but the main general purpose meta-heuristic algorithms are:

- Genetic Algorithm (GA) [21],
- Differential Evolution [22],
- Ant Colony [23],
- Bee Colony [24] and
- Particle Swarm Optimization (PSO) [8]

Between all these algorithms, the GA is the most studied method for scheduling problems. Although, recent studies demonstrate that, for other problems, the PSO has faster convergence and similar results to the GA [10] [25] [26] [27] [28]. As this work focus on solving the maintenance schedule in a shorter time, it will focus on the PSO algorithm.

A. CPSO

Originally, the PSO movements the particles in a continuous independent n dimensional search space. Although, in the combinatorial model, CPSO, it's not possible to have the same value in two different vector positions. Therefore, a new position actualization process is necessary to this model.

As described in [10], it's not necessary modify the velocity equation, only the position equation: $x(t+1) = x(t)+v(t+1)$. This operation is modified to a 5-step process illustrated on Figure 1.

The first step is to limit the velocity between $-N$ and N, where N is the vector length. Then, normalize the vector by dividing it by N. Using this normalized velocity vector as a probability to select a target position, some positions are randomly chosen to the next two steps.

A selected vector position is compared with the target particle. The original vector chosen position task will be named T_A and the target vector chosen position task as T_B. The original position vector is evaluated to test if the switch between T_A and T_B won't break any precedence condition. If T_A can switch with positions with T_B on the original vector, then they change places. This operation is repeated for all the selected positions, but a switched tasks should be skipped. The final vector is considered the actualized position vector.

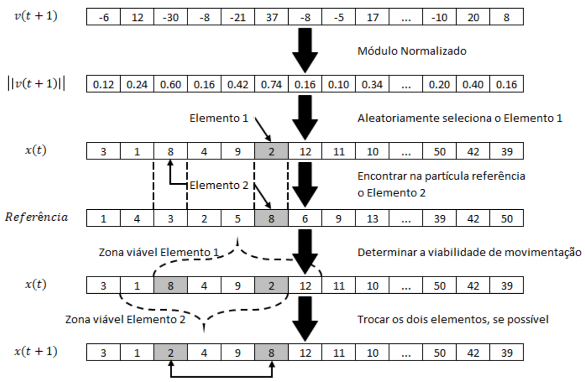

Fig. 1. Exemplo de atualização do PSO Combinatorial

B. PSO+

When the CPSO is a modification on the original PSO to handle combinatorial problems, the PSO+ [9] is a variant that is specialized in handling complex restrictions. The main components responsible to this capability are:

- Movement inertia;
- Invalid solution corrector;
- Exploration movement.

To that components to work, the original search swarm is modified in three swarms that, together, explore the solution space:

- Frontier swarm
- Reference swarm
- Support swarm

The frontier swarm is responsible to disperse in the search space looking for the best solution. When a particle in this swarm has a valid position, it can evolve similar to the original PSO or do a exploration movement, move in the direction of the support swarm to better explore the space. Although, when it‘s position is invalid, the algorithms tries to correct the particle evaluation with the reference swarm help.

The support swarm is responsible to keep invalid positions found during the initialization to push the frontier swarm towards the limits between the valid and invalid space, helping the search on the restriction limit. This process occurs replacing the global target on the original PSO movement to the position of a support particle, redirecting the valid frontier particle actualization.

On the other hand, the reference swarm is responsible to store the best evaluated valid positions. Based on GA solution corrector, [29], [30], When a invalid frontier particle need to be evaluated, a random reference particle is chosen. This valid particle is used as target to a virtual movement between the frontier particle and the reference particle. If a new valid position is found, the invalid particle position is not modified, but it‘s evaluation is considered the new valid position evaluation. If this evaluation is better then the reference particle evaluation, the old reference particle is replaced with this new position.

V. CPSO+

This study proposes a new PSO variant, the CPSO+, which unifies the order modifications from the CPSO, with the restriction handle compenets from the PSO+.

Fig. 2. Elementos do CPSO+

A. Worker decoder

It’s important to highlight that the resources allocated on a Flexible Shop Scheduling Problem can be either raw materials for a process, tools to execute the task, or workers with the skill to complete the job. In this last case, not all employees are able to execute all activities, being necessary a matrix to correlate the resources with the possible activities. Even thought there are multiple workers combinations for one job and multiple jobs possible for a worker, in this kind of problem, one worker can only execute one job at a time and only one worker will be necessary for each task.

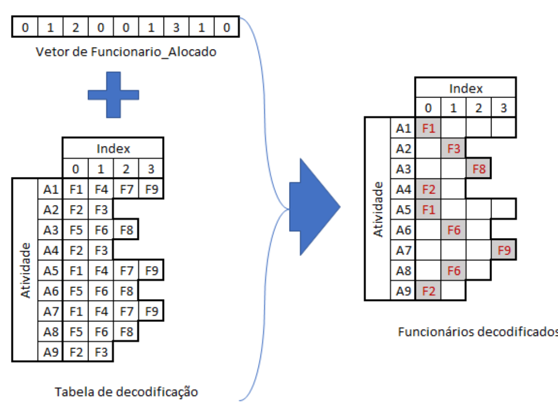

Fig. 3. Exemplo de representação de seleção de recurso

With those restrictions, it’s possible to develop a worker selection vector to model which employee will work on each scheduled task. In this model, illustrated on Figure 3, the N^{th} position on the vector Allocated_worker represents the N^{th} task. On the other hand, the value in the vector position, **index**, represents which column on the Decodification_table will be selected for this task. The position-index combination, then, is decrypt in the selected worker for the task.

Those construction allows the optimization algorithm to search on a continuous integer space that can be uniquely converted on a feasible worker. These search space is simpler to optimize since all restrictions are treated by the Decodification_table, improving the method speed.

B. CPSO+ position update

To use the restriction handle components from the PSO+, the CPSO+ has to adapt the orginal position update formula from each of the three swarms.

First, the solution corrector movement is modified from a random speed CPSO position update with a reference swarm particle as target. For the exploration movement, the target particle is a support swarm particle and the speed is the frontier particle current speed. Finally, for the normal movement, the frontier particle randomly chooses between the best past position and the global best position as target for the movement. After the position update, each particle is evaluated by the order decoder and the worker decoder an judged valid or invalid.

C. CPSO+ with modified initialization

As the CPSO+ has three different swarms, the initialization process takes more time then the CPSO. To minimize this time invest in creating the swarms that handle the restrictions, the reference swarm and the support swarm, a new initialization method is purposed.

While the original initialization method searchs for N valid particles for the reference swarm, N invalid particles for the support swarm and N random particles for the frontier swarm, the $CPSO{+}_m$, CPSO+ modified for faster initialization, only generates $2N$ particles. Each created particle is evaluated as valid or invalid and, in a balanced search space, around half the particles should be valid and the other half invalid. Although, in a unbalanced space is more likely to not found any valid particles, if mostly invalid, or invalid, if mostly valid.

In those cases, the CPSO+ will start to search for a solution without the restriction handling swarms saving time searching for a specific type of particle that may not even exist on a specific problem. After a fill eras, when it founds the specific particle type it populates the extra swarms and start to use them to better explore the search space. On the other hand, as it does not have the other swarms, this variation can present worst results then the long initialization version since it moves similar to a simple CPSO at the initial eras.

VI. Simulations & Results

To compare the literature model, CPSO, with the model purposed on this study, , CPSO+ and $CPSO{+}_m$, four simulation scenarios were run:

- a complete invalid search space
- a complete valid search space
- a predominant valid search space
- a predominant invalid search space

Those conditions were selected to evaluate the new algorithms in all possible applicable applications. As the space viability is determined only by the jobs deadline, all simulations are composed by the same tasks, workers and boarding dates and exclusively different by the deadline of one job. The shared parameters are exhibited in Table I, while the deadline of the discriminant job is displayed in Table II.

TABLE I
Shared simulation parameters

Parameter	Value
Distinct needed skills	7 skills
Number of employees	6 employees for each skill (total = 42)
Boarding limit	4
Number of activities	Job 01 - 43 Job 02 - 42 Job 03 - 53 Job 04 - 75 Total - 213
Deadline	Job 01 - No Deadline Job 02 - Change with simulation case Job 03 - No Deadline Job 04 - No Deadline
Fixed boarding days	0 ; 6 ; 13 ; 20 ; 27 ; 34 ; 41 ; 48 ; 55 ; 62 ; 69 ; 76 ; 83
Max schedule length	93 days

TABLE II
Job 02 deadline per simulation scenario

Scenario	Job 02 deadline
Scenario 01	Day 2
Scenario 02	No deadline
Scenario 03	Day 70
Scenario 04	Day 15

Scenario 01 presents an impossible short deadline, to evaluate how each model handles non convergent searches. In Scenario 02, there is no deadline for the job, to evaluate how the proposed models, CPOS+ and $CPSO{+}_m$, perform without the Support swarm. Scenario 03 represents a loose deadline, to evaluate the convergence time in a simple non-linear restriction scenario. In the last scenario, Scenario 04, a strict deadline is provided, to evaluate the convergence time and results on a more common industrial scenario.

Each model, CPSO; CPSO+; $CPSO{+}_m$, was tested in each scenario twenty times to compare the final results dispersion, the average initialization time and the average search time. The results found where disposed on Tables III, V and IV.

TABLE III
Solution quality results

Avg MAX	CPSO	CPSO+	$CPSO{+}_m$
Scenario 01	-8900	-8980	-8945
Scenario 02	-1080	-1095	-1080
Scenario 03	-1415	-1055	-1140
Scenario 04	-7650	-7175	-7285

Comparing the results obtained, Scenario 01 and Scenario 02 demonstrate that the proposed models, CPSO+ and $CPSO + m$, provide similar results to the CPSO, but with longer search times, which allows to conclude that those new

TABLE IV
INITIALIZATION TIME RESULTS

Avg Initialization time	CPSO	CPSO+	$CPSO{+}_m$
Scenario 01	18 s	1 min 14 s	36 s
Scenario 02	17 s	1 min 7 s	29 s
Scenario 03	18 s	1 min 13 s	36 s
Scenario 04	19 s	1 min 14 s	36 s

TABLE V
SEARCH TIME RESULTS

Avg search time	CPSO	CPSO+	$CPSO{+}_m$
Scenario 01	2 min 10 s	3 min 36 s	2 min 45 s
Scenario 02	1 min 17 s	2 min 32 s	1 min 55 s
Scenario 03	2 min 24 s	7 min 31 s	9 min 40 s
Scenario 04	2 min 32 s	3 min 11 s	3 min 25 s

models are not good for non restricted applications. On the other hand, in Scenario 03 and Scenario 04, with access to all three swarms, the proposed models present better results then the CPSO with a reasonable search time increase for the offshore industry application.

Even though the $CPSO + m$ presents similar search time to the CPSO+ in Scenario 03, the first demonstrates similar results to the second with shorter initialization time. Therefore, it is possible to argue that the $CPSO{+}m$ has a better balance between result quality and total search time to be applicable on the offshore maintenance scheduling segment then the CPSO+ and the CPSO.

VII. CONCLUSION

This study, with the objective of develop a decision support system to the offshore scheduling maintenance team, describes a new combinatorial optimization model by particle swarm: the CPSO+. The purposed model is an join between the PSO+ and the CPSO to solve the complex problem of finding the best maintenance task and worker sequence that maximizes the company profit in the first three months, respecting the estimated deadline and sector restrictions.

The CPSO+ and it's variant, $CPSO{+}_m$, performance were evaluated comparing their results whith the CPSO results. These three models where tested on four simulations with only one free parameter, the jobs deadline. The simulations conditions were: a) impossible deadline; b) no deadlines; c) a job with a distant deadline and d) a job with a short deadline.

All evaluated models were able to solve the described simulations with different processing speeds and result quality. It's important to highlight the $CPSO{+}_m$ results, that presents a balanced result between processing time, 4 min and 25 sec, and solution quality.

REFERENCES

[1] G. Miragliotta, A. Sianesi, E. Convertini, and R. Distante, "Data driven management in industry 4.0: a method to measure data productivity," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 19 – 24, 2018, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896318313521

[2] H. Lu, J. Shi, Z. Fei, Q. Zhou, and K. Mao, "Analysis of the similarities and differences of job-based scheduling problems," *European Journal of Operational Research*, vol. 270, no. 3, pp. 809 – 825, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221718300870

[3] M. F. Tompkins, "Optimization techniques for task allocation and scheduling in distributed multi-agent operations," Dissertação de Mestrado, Massachusetts Institute of Technology, 2003.

[4] P. Balaprakash, V. Morozov, R. Kettimuthu, K. Kumaran, and I. Foster, "Improving data transfer throughput with direct search optimization," in *2016 45th International Conference on Parallel Processing (ICPP)*, Aug 2016, pp. 248–257.

[5] S. Lendl, A. Ćustić, and A. P. Punnen, "Combinatorial optimization with interaction costs: Complexity and solvable cases," *Discrete Optimization*, vol. 33, pp. 101 – 117, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1572528618300434

[6] M. Goerigk and S. J. Maher, "Generating hard instances for robust combinatorial optimization," *European Journal of Operational Research*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221719306009

[7] J. Gajarský, P. Hliněný, M. Koutecký, and S. Onn, "Parameterized shifted combinatorial optimization," *Journal of Computer and System Sciences*, vol. 99, pp. 53 – 71, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000018306986

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[9] M. R. Kohler, "Pso+: Algoritmo com base em enxame de partículas para problemas com restrições lineares e não lineares," Tese de Doutorado, PUC-RIO, 2017.

[10] H. Zhang, X. Li, H. Li, and F. Huang, "Particle swarm optimization-based schemes for resource-constrained project scheduling," *Automation in Construction*, vol. 14, no. 3, pp. 393 – 404, 2005, international Conference for Construction Information Technology 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580504001128

[11] C. M. d. FREITAS, C. A. V. d. SOUZA, J. M. H. MACHADO, and M. F. d. S. PORTO, "Acidentes de trabalho em plataformas de petróleo da bacia de campos," *Caderno de Saúde Pública*, vol. 17, no. 1, pp. 117–130, 2001. [Online]. Available: https://doi.org/10.1590/S0102-311X2001000100012

[12] S. M. R. MARTINS, "O trabalho offshore: Um estudo sobre as repercussões do confinamento nos trabalhadores das plataformas de petróleo na bacia de campos, rj," Dissertação de Mestrado, UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE, 2006.

[13] L. Shen, S. Dauzère-Pérès, and J. S. Neufeld, "Solving the flexible job shop scheduling problem with sequence-dependent setup times," *European Journal of Operational Research*, vol. 265, no. 2, pp. 503 – 516, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037722171730752X

[14] G. Gong, Q. Deng, X. Gong, W. Liu, and Q. Ren, "A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators," *Journal of Cleaner Production*, vol. 174, pp. 560 – 576, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959652617324952

[15] M. Dai, D. Tang, A. Giret, and M. A. Salido, "Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 143 – 157, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584518305222

[16] L. Fanjul-Peyro, R. Ruiz, and F. Perea, "Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times," *Computers & Operations Research*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054818301916

[17] K. Fleszar and K. S. Hindi, "Algorithms for the unrelated parallel machine scheduling problem with a resource constraint," *European Journal of Operational Research*, vol. 271, no. 3, pp. 839 – 848, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221718304752

[18] M. R. GAREY, "Computers and intractability : A guide to the theory of np-completeness, freeman," *Fundamental*, 1997. [Online]. Available: https://ci.nii.ac.jp/naid/10010426106/en/

[19] A. Zhang, Y. Chen, L. Chen, and G. Chen, "On the np-hardness of scheduling with time restrictions," *Discrete*

Optimization, vol. 28, pp. 54 – 62, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1572528616301396

[20] J. M. Balera and V. A. de Santiago Júnior, "A systematic mapping addressing hyper-heuristics within search-based software testing," *Information and Software Technology*, vol. 114, pp. 176 – 189, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584919301430

[21] S. Yoo and M. Harman, "Pareto efficient multi-objective test case selection," in *Proceedings of the 2007 International Symposium on Software Testing and Analysis*, ser. ISSTA '07. New York, NY, USA: ACM, 2007, pp. 140–150. [Online]. Available: http://doi.acm.org/10.1145/1273463.1273483

[22] D. Vargas, A. Lemonge, H. Barbosa, and H. Bernardino, "Um algoritmo baseado em evolução diferencial para problemas de otimização estrutural multiobjetivo com restrições," *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, vol. 32, no. 2, pp. 91 – 99, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0213131515000231

[23] T. SELÇUK and A. ALKAN, "Detection of microaneurysms using ant colony algorithm in the early diagnosis of diabetic retinopathy," *Medical Hypotheses*, vol. 129, p. 109242, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306987719304098

[24] M. Zhang, Y. Tan, J. Zhu, Y. Chen, and H. Liu, "Modeling and simulation of improved artificial bee colony algorithm with data-driven optimization," *Simulation Modelling Practice and Theory*, vol. 93, pp. 305 – 321, 2019, modeling and Simulation of Cloud Computing and Big Data. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X1830087X

[25] A. Arram and M. Ayob, "A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems," *Computers & Industrial Engineering*, vol. 133, pp. 267 – 274, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835219302773

[26] F. Guijarro, M. Martínez-Gómez, and D. Visbal-Cadavid, "A combined genetic algorithm and inverse data envelopment analysis model for target setting in mergers," *Expert Systems with Applications: X*, p. 100012, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2590188519300125

[27] H. Che and J. Wang, "A collaborative neurodynamic approach to global and combinatorial optimization," *Neural Networks*, vol. 114, pp. 15 – 27, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608019300425

[28] M. Hamdi and M. Zaied, "Resource allocation based on hybrid genetic algorithm and particle swarm optimization for d2d multicast communications," *Applied Soft Computing*, vol. 83, p. 105605, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494619303850

[29] Z. Michalewicz and C. Z. Janikow, "Genocop: A genetic algorithm for numerical optimization problems with linear constraints," *Commun. ACM*, vol. 39, no. 12es, Dec. 1996. [Online]. Available: http://doi.acm.org/10.1145/272682.272711

[30] Z. Michalewicz and G. Nazhiyath, "Genocop iii: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints," in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, vol. 2, Nov 1995, pp. 647–651 vol.2.