Robotic Flow Shop Scheduling with Parallel Machines and No-Wait Constraints in an Aluminium Anodising Plant with the CMAES Algorithm

Carina M. Behr and Jacomine Grobler(✉)

Department of Industrial and Systems Engineering,
University of Pretoria, Pretoria, South Africa
jacomine.grobler@gmail.com

Abstract. This paper proposes a covariance matrix adaptation evolution strategy (CMAES) based algorithm for a robotic flow shop scheduling problem with multiple robots and parallel machines. The algorithm is compared to three popular scheduling rules as well as existing schedules at a South African anodising plant. The CMAES algorithm statistically significantly outperformed all other algorithms for the size of problems currently scheduled by the anodising plant. A sensitivity analysis was also conducted on the number of tanks required at critical stages in the process to determine the effectiveness of the CMAES algorithm in assisting the anodising plant to make business decisions.

Keywords: Robotic flow shop scheduling
Covariance matrix adaptation evolution strategy

1 Introduction

Anodising is an electrolytic reaction used to produce a layer of aluminium oxide on an aluminium alloy. Anodising production lines often consist of a series of chemical processes with material needing to be moved from station to station by means of overhead cranes - a complex production scheduling problem. Developing an optimization algorithm to solve a problem of this nature has both practical and academic significance. Firstly, effective production scheduling has a major impact on the cost of production, utilization of resources and customer satisfaction. Secondly, the optimization problem is a discrete-valued optimization problem with a number of complicated constraints, which is challenging to solve by means of an evolutionary algorithm.

In this paper, a scheduling problem at a South African anodising plant is formulated as a robotic flow shop problem with multiple robots and parallel machines with no-wait constraints. A covariance matrix adaptation evolution

L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10841, pp. 302–312, 2018.
https://doi.org/10.1007/978-3-319-91253-0_29

strategy (CMAES) [1] based scheduling algorithm is developed to solve the problem. The CMAES based algorithm is benchmarked against a number of scheduling heuristics on real data from the anodising plant. The results are also compared against the existing schedules generated by the production personnel. The CMAES algorithm statistically significantly outperformed all the benchmark algorithms and existing schedules for problems where around 35 jobs needed to be scheduled on 13 stations. As problem size increased the performance of the CMAES algorithm worsened. A sensitivity analysis was also conducted on the number of tanks required at critical stages in the process.

This paper is significant because, to the best of the authors' knowledge, it describes the first CMAES based algorithm for solving a robotic flow shop problem.

The rest of the paper is organized as follows: Sect. 2 describes the actual scheduling problem in more detail and provides an overview of existing approaches used in literature for solving similar problems. Section 3 provides a brief introduction to CMAES and describes the scheduling algorithm in more detail. The experimental setup and results are described in Sect. 4 before the CMAES as tool for business decision making is evaluated in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Robotic Flow Shop Problems

The robotic flow shop problem with parallel processing stations and multiple robots requires n jobs to be processed by s stations, one operation on each station. All jobs complete processing in the same sequence with standard processing times [2]. The purpose is to determine the sequence in which the jobs should be processed to minimize the makespan (total time to complete all required jobs). The cranes in the anodising process (modelled as robots) have machine availability constraints, which imply that a crane can only be utilized to move material between two stations if it is not already in use for another move. A move time is calculated for each job to be transported between two stations as the time required for the crane to move to the pickup location and the actual time required for the crane to move the job to the next station.

Figure 1 illustrates the specific anodising process and its stations. The brackets indicate parallel stations. The hot etch, anodising and sealing stations are critical stations or stations with a no-wait scenario, which means the cranes need to pick up jobs from these stations the moment the processing time has been completed (no-wait constraints). The remaining stations may wait for the crane to become available before a pickup is made.

Since the cranes cannot cross over each other, the two cranes are each preallocated stations to service with the first crane servicing the first half of the stations and the second crane the second half of the stations. A detailed formulation of this problem can be found in [3] and will thus not be repeated here for the sake of conciseness.

A number of researchers have already solved variations of the robotic flow shop scheduling problem. The first variations focused largely on solving problems

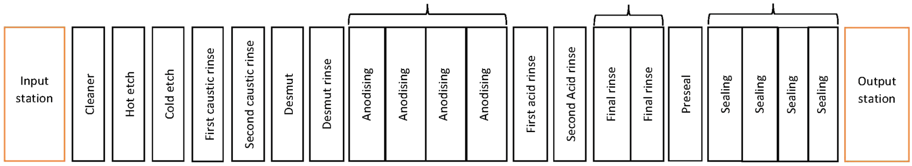

Fig. 1. Anodising stations.

where only one robot was available to move products between stations [4,5]. The problems were soon expanded to consider multiple robots [6,7] and parallel machines [3,8], where a stage has more than one machine which can be used to process a job.

Three types of part pickup criteria have been considered, namely no-wait [9, 10], interval [11,12], and free pickup. Additional complexities such as reentrance [13], fixed robot routes [14] and sequence-dependent setup-times [4] can also be found.

With regard to solution strategies, formulating problems as mixed integer linear programmes which are solved exactly by software packages such as CPLEX [3,6,11,13], is common for small sized problems. Larger problems are solved by heuristics [8,14] and meta-heuristics such as genetic algorithms [5], ant colony optimization [16] and cuckoo search [4].

Two main conclusions can be reached from analyzing the existing robotic flow shop literature. Firstly, there is no evidence of previous use of CMAES for solving a robotic flow shop scheduling problem. Secondly, the number of jobs typically considered in literature are significantly less than the number of jobs which need to be considered for the anodising plant described in this paper. Zhou et al. [11], for example, focused on scheduling 3 jobs on 24 stages (resulting in 72 operations) and Li et al. [3] focused on scheduling 3 jobs on 20 stages (resulting in 60 operations). The problems considered in this paper ranges from 455 operations to 1300 operations. The next section will discuss the CMAES based algorithm developed to solve these problems in more detail.

3 CMAES for Robotic Flow Shop Scheduling

Lei and Wang [15] proved that robotic flow shop problems with time window constraints, such as no-wait constraints, are np-complete. The more complicated problem addressed in this paper can thus not be solved to optimality in polynomial time. Approximation algorithms such as heuristic rules or evolutionary algorithms are thus logical options. CMAES was selected as basis for the scheduling algorithm due to its excellent performance versus more well known evolutionary algorithms such as genetic algorithms (GAs) and differential evolution algorithms on continuous optimization problems [1]. Furthermore, CMAES has only one parameter to tune and requires significantly fewer individuals per iteration compared with, for example a GA, to obtain satisfactory performance.

3.1 Background on CMAES

CMAES is a stochastic, non-linear optimization algorithm. The CMAES algorithm consists of four main phases, namely solution generation, selection and recombination, covariance matrix update, and step size update. During the first generation phase, a population of solutions is generated at each iteration according to a multivariate normal distribution such that

$$\boldsymbol{x}_i(t+1) \sim N(\boldsymbol{m}(t), \sigma^2_{CMA}(t))\boldsymbol{C}(t), \tag{1}$$

where $\boldsymbol{x}_i(t+1)$ is the i^{th} candidate solution at iteration $t+1$, $N(\boldsymbol{m}(t), \sigma^2_{CMA}(t))$ denotes a normal distribution with mean $\boldsymbol{m}(t)$ and standard deviation $\sigma_{CMA}(t)$. The mean of the CMAES population at time t is denoted by $\boldsymbol{m}(t)$, σ_{CMA} denotes the step size of the algorithm at time t, and $\boldsymbol{C}(t)$ is the covariance matrix at time t. After the solutions are evaluated and sorted, selection and recombination takes place by adjusting the mean of the population as follows:

$$\boldsymbol{m}(t+1) = \sum_{k=1}^{n_s} w_k \boldsymbol{x}_k, \tag{2}$$

where n_s is the population size and w_k is the k^{th} recombination weight in the CMAES algorithm.

The covariance matrix, $\boldsymbol{C}(t)$, is then updated as:

$$\begin{aligned}\boldsymbol{C}(t+1) = (1-c_{cov})\boldsymbol{C}(t) + \frac{c_{cov}}{\mu_{cov}} p_{c_{CMA}} p^T_{c_{CMA}} + c_{cov}\left(1 - \frac{1}{\mu_{cov}}\right) \\ \times \sum_{k=1}^{n_s} w_k \left(\frac{x_k(t+1) - \boldsymbol{m}(t)}{\sigma_{CMA}(t)}\right)\left(\frac{x_k(t+1) - \boldsymbol{m}(t)}{\sigma_{CMA}(t)}\right)^T,\end{aligned} \tag{3}$$

where

$$\mu_{cov} \geq 1, \tag{4}$$

$$\mu_{cov} = \mu_{eff}, \text{and} \tag{5}$$

$$c_{cov} \approx \min(\mu_{cov}, \mu_{eff}, n_x^2)/n_x^2. \tag{6}$$

The symbol c_{cov} denotes the learning rate for the covariance matrix update, μ_{eff} denotes the variance effective selection mass and μ_{cov} denotes the parameter which weighs between the rank-one update and rank-μ update. The rank-one update uses only the previous iteration to estimate the covariance matrix where the rank-μ update uses all previous iterations. n_x is the number of problem dimensions.

The CMAES algorithm makes use of cumulative step-size adaptation. A cumulative path is used which is a combination of all the steps an algorithm has made with the importance of a step decreasing exponentially with time [17]. Two evolution paths are used in the CMAES algorithm, the anisotropic evolution path, $p_{c_{CMA}}$, associated with the covariance matrix and the isotropic evolution path, p_σ, associated with the step size. $p_{c_{CMA}}$ is calculated as follows:

$$p_{c_{CMA}} = (1 - c_{c_{CMA}})p_{c_{CMA}} + \sqrt{c_{c_{CMA}}(2 - c_{c_{CMA}})\mu_{eff}} \left(\frac{\boldsymbol{m}(t+1) - \boldsymbol{m}(t)}{\sigma_{CMA}(t)} \right), \tag{7}$$

where μ_{eff} is given by

$$\mu_{eff} = \left(\sum_{k=1}^{n_s} w_k^2 \right)^{-1} \tag{8}$$

and $c_{c_{CMA}}$ is the backward time horizon of the anisotropic evolution path.

Finally, the step size, $\sigma_{CMA}(t+1)$, is updated as follows:

$$\sigma_{CMA}(t+1) = \sigma_{CMA}(t) \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|p_\sigma(t+1)\|}{E\|N(\mathbf{0}, \boldsymbol{I})\|} - 1 \right) \right), \tag{9}$$

where d_σ is the damping parameter in the CMAES algorithm, $\frac{1}{c_\sigma}$ is the backward time horizon of the isotropic evolution path, p_σ:

$$p_\sigma = (1 - c_\sigma)p_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}}\boldsymbol{C}(t)^{-0.5} \left(\frac{\boldsymbol{m}(t+1) - \boldsymbol{m}(t)}{\sigma_{CMA}(t)} \right). \tag{10}$$

The Covariance Matrix Adaption Evolution Strategy (CMAES) algorithm has already been used for a number of real world problems such as to optimise the layout of a three column wind farm [18] and for the optimization of irrigation scheduling [19].

3.2 The CMAES-Based Robotic Flow Shop Scheduling Algorithm

Since the CMAES algorithm operates in a continuous space, a mapping mechanism is required to convert the candidate solutions ($\boldsymbol{x}_i(t+1)$) to valid schedules [20]. Each candidate solution is interpreted as a set of priorities for the jobs to be scheduled. This prioritized list of jobs is then provided as input to a scheduling algorithm described in Fig. 2. The algorithm provides the makespan associated with the schedule of each candidate solution which is then returned to the CMAES algorithm.

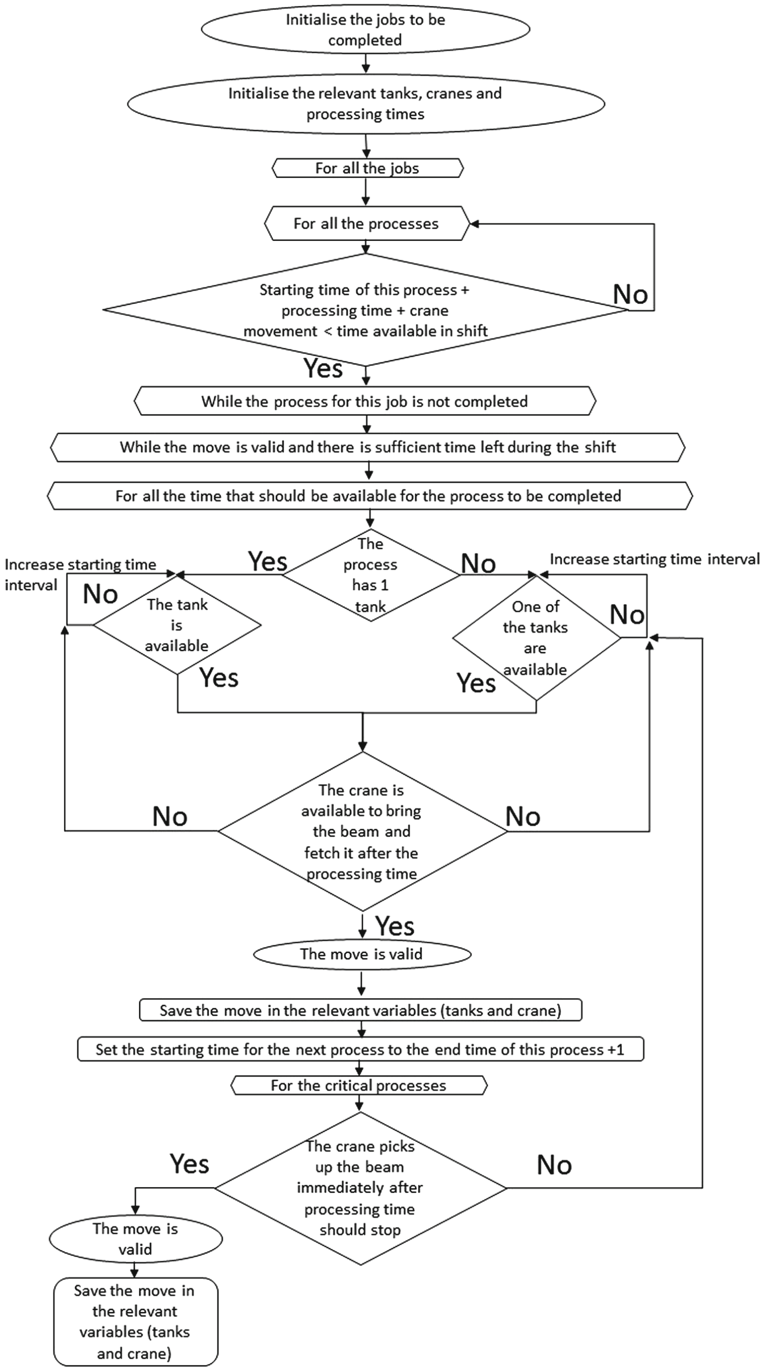

Fig. 2. Flowchart of the algorithm used to calculate the fitness function of each individual.

4 Empirical Evaluation

The evaluation of the CMAES scheduling algorithm was conducted on 5 datasets derived from real world data from the South African anodising plant. The datasets range in size from 34 jobs (corresponding to the current planning horizon) to 100 jobs which needs to be produced on 13 processing stages. These datasets are available for comparison purposes from the corresponding author.

The parameter settings used for the CMAES algorithm is based on recommendations from literature [1] and is listed in Table 1.

Table 1. The CMAES algorithm parameters.

Parameter	Value used
Maximum iterations	$\frac{1e3(n_x+5)^2}{\sqrt{n_s}}$
Search interval ($[LB, UB]$)	$[-100, 100]$
Population size (n_s)	$4 + floor(3 \log n_x)$
Sigma	$0.35(UB - LB)$

For the first three datasets, the actual production schedule as planned and executed by the anodising plant could be obtained. These schedules, referred to as the "as-is" schedules can be used as a baseline to compare the performance of the CMAES algorithm. Furthermore, another three production scheduling rules were used for benchmarking purposes. These rules are the current state-of-the-art algorithms developed specifically for the anodising plant's problem. The rules are FIFO (First In, First Out), Priority (where priority orders are completed first and then the as-is schedule from the plant is followed for the rest of the orders) and SPT (shortest processing time) (all 10 µm orders, followed by 15 µm and then 25 µm).

The results of the comparison between the CMAES scheduling algorithm, the benchmarking algorithms and the as-is schedules are recorded in Table 2. Due to the stochastic nature of the CMAES algorithm, all CMAES results were recorded over 30 independent simulation runs. Throughout the rest of this section, μ and σ respectively denote the mean and standard deviation associated with the makespan of the algorithm.

A statistical analysis was conducted to validate the results obtained. For every dataset, a Mann-Whitney U test at 95% significance was performed to compare the CMAES algorithm results to the benchmarking algorithms. The results in bold indicate that a statistical significant performance improvement was obtained. As can be seen, the CMAES algorithm outperforms the other algorithms for the first three datasets. Up to 115 min can be saved per day by the anodising plant if they utilize the CMAES algorithm instead of their existing scheduling procedures. With regard to the larger two datasets, the CMAES performs statistically similar to the SPT heuristic, indicating that further design

Table 2. Comparative results of the CMAES scheduling algorithm to the benchmark algorithms and as-is schedules.

Dataset	Operations	CMAES			FIFO	SPT	Priority	As-is
		μ	σ	Time (s)	C	C	C	C
07-Aug	34	**457.23**	11.596	412.16	603	571	600	572
22-Aug	34	**470.6**	13.051	415.72	556	546	579	560
23-Aug	42	**544.3**	10.639	649.57	643	608	727	658
50	50	627	0	2276.28	NA	627	NA	798
100	100	1177	0	5985.65	NA	1177	NA	1621

improvements will need to be made to the CMAES algorithm if a larger scheduling horizon is considered in future.

The time required to obtain a solution for each problem was also recorded in Table 2. A computer running Windows 7 Enterprise with an Intel(R) Core(TM) i7-4710MQ CPU @ 2.50 GHz with 8 Gb RAM was used to obtain the results. Running an algorithm overnight to schedule the next morning's production is considered acceptable in industry and thus the computational time required to obtain a solution is not an issue.

5 Using the CMAES Algorithm for Business Decisions

From the results obtained, it is clear that the CMAES algorithm could add value by reducing the total time required to produce material. This section investigates the use of the CMAES algorithm to assist the anodising plant in making business decisions. A sensitivity analysis was conducted on the number of tanks used at two critical stages in the manufacturing process. Three scenarios were developed. Scenario 1 involved the addition of two anodising tanks to the existing four tanks at the anodising station. Scenario 2 involved adding two sealing tanks to the existing four sealing tanks at the sealing station. Scenario 3 involved adding both additional anodising and sealing tanks to the existing production line. The CMAES algorithm was used to solve the three scenarios and the results were recorded in Table 3.

A statistical analysis was again conducted. For every comparison of scenarios, a Mann-Whitney U test was performed (using the two sets of 30 data points of the two scenarios being compared) and if the existing scenario statistically significantly outperformed the second scenario, a win was recorded. If no statistical difference could be observed a draw was recorded. If the second scenario resulted in statistically significantly better results than the as-is scenario, a loss was recorded for the as-is scenario. As an example, (5-0-0) in row 1 column 1, indicates that the "as-is" scenario significantly outperformed Scenario 1 for five of the datasets. No draws or losses were recorded (Table 4).

Interestingly, the results show that adding additional tanks does not have a significant impact on the schedule. One possible explanation could be that the

Table 3. Sensitivity analysis of the CMAES algorithm results to additional tanks.

Dataset	As-is setup		Scenario 1		Scenario 2		Scenario 3	
	μ	σ	μ	σ	μ	σ	μ	σ
07-Aug	457.23	11.596	524.23	1.2507	536.23	11.227	458.03	12.076
22-Aug	470.6	13.051	496.93	4.0338	507.37	14.39	451	0
23-Aug	544.3	10.639	628.73	0.82768	648.53	5.8648	545.23	11.596
50	627	0	630.37	3.81	649.9	3.2308	627	0
100	1177	0	1182.5	5.0596	1243.3	4.4347	1177	0

Table 4. Hypotheses analysis of the impact of additional tanks on scheduling performance.

Scenario 1	Scenario 2	Scenario 3	TOTAL
5-0-0	5-0-0	0-4-1	10-4-1

process bottleneck is now simply moved to another resource, such as the cranes, which are required to move all jobs between stations. Future work can focus on a more in-depth analysis of the system bottlenecks.

6 Conclusion

This paper described the development of a CMAES based algorithm for a robotic flow shop scheduling problem with multiple robots and parallel machines. Data from a South African anodising plant was used to compare the CMAES algorithm to three popular scheduling rules as well as existing schedules used at the plant. The CMAES algorithm statistically significantly outperformed all other algorithms for problems with around 500 operations. An example analysis was also conducted to show how the algorithm can be used to evaluate various system parameters.

Future research opportunities lie in the improvement of algorithm performance on larger problems, benchmarking the CMAES algorithm against other meta-heuristic algorithms, more in-depth tuning of the CMAES parameters, and a more thorough analysis of the system bottlenecks.

Acknowledgements. This work is based on the research supported wholly or in part by the National Research Foundation of South Africa (Grant Number 109273). The authors would also like to thank the University of Twente for their financial support.

References

1. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, pp. 1769–1776 (2005)
2. Tang, L.X., Liu, P.: Two-machine flow shop scheduling problems involving a batching machine with transportation or deterioration consideration. Appl. Math. Model. **33**, 1187–1199 (2009)
3. Li, X., Chan, F.T.S., Chung, S.H.: Optimal multi-degree scheduling of multiple robots without overlapping in robotic flow shops with parallel machines. J. Manufact. Syst. **36**, 62–75 (2015)
4. Majumder, A., Laha, D.: A new cuckoo search algorithm for 2-machine robotic cell scheduling problem with sequence-dependent setup times. Swarm Evol. Comput. **28**, 131–143 (2016)
5. Lim, J.M.: A genetic algorithm for a single hoist scheduling in the printed-circuit-board electroplating line. Comput. Ind. Eng. **33**(3–4), 789–792 (1997)
6. Che, A., Lei, W., Feng, J., Chu, C.: An improved mixed integer programming approach for multi-hoist cyclic scheduling problem. IEEE Trans. Autom. Sci. Eng. **11**(1), 302–309 (2014)
7. Che, A., Chu, C.: Optimal scheduling of material handling devices in a PCB production line: problem formulation and a polynomial algorithm. Math. Probl. Eng. **2008** (2008). Article ID 364279
8. Geismar, H.N., Pinedo, M., Sriskandarajah, C.: Robotic cells with parallel machines and multiple dual gripper robots: a comparative overview. IIE Trans. **40**(12), 1211–1227 (2008)
9. Che, A., Chu, C.: Multi-degree cyclic scheduling of a no-wait robotic cell with multiple robots. Eur. J. Oper. Res. **199**(1), 77–88 (2009)
10. Che, A., Chabrol, M., Gourgand, M., Wang, Y.: Scheduling multiple robots in a no-wait re-entrant robotic flow shop. Int. J. Prod. Econ. **135**(1), 199–208 (2012)
11. Zhou, Z., Che, A., Yan, P.: A mixed integer programming approach for multi-cyclic robotic flow shop scheduling with time window constraints. Appl. Math. Model. **36**(8), 3621–3629 (2012)
12. Dawande, M., Geismar, H.N., Pinedo, M., Sriskandarajah, C.: Throughput optimization in dual-gripper interval robotic cells. IIE Trans. **42**(1), 1–15 (2009)
13. Li, X., Fung, R.Y.: A mixed integer linear programming solution for single hoist multi-degree cyclic scheduling with reentrance. Eng. Optim. **46**(5), 704–723 (2014)
14. Kats, V., Levner, E.: Parametric algorithms for 2-cyclic robot scheduling with interval processing times. J. Sched. **14**(3), 267–279 (2011)
15. Lei, L., Wang. T.: A proof: the cyclic hoist scheduling problem is NP-complete. Graduate School of Management, Rutgers University, Working Paper, pp. 89–116 (1989)
16. Elmi, A., Topaloglu, S.: Cyclic job shop robotic cell scheduling problem: ant colony optimization. Comput. Ind. Eng. **111**, 417–432 (2017)
17. Chotard, A., Auger, A., Hansen, N.: Cumulative step-size adaptation on linear functions. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012. LNCS, vol. 7491, pp. 72–81. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32937-1_8
18. Saravanan, A.J., Karthikeyan, C.P., Samuel, A.A.: Optimization of exogenous and endogenous variables for a three column wind farm using CMAES. Appl. Mech. Mater. **573**, 777–782 (2014)

19. Belaqziz, S., Mangiarotti, S., Le Page, M., Khabba, S., Er-Raki, S., Agouti, T., Drapeau, L., Kharrou, M.H., El Adnani, M., Jarlan, L.: Irrigation scheduling of a classical gravity network based on the covariance matrix adaptation - evolutionary strategy algorithm. Comput. Electron. Agric. **102**, 64–72 (2014)
20. Grobler, J., Engelbrecht, A.P., Kok, S., Yadavalli, V.S.S.: Metaheuristics for the multi-objective FJSP with sequence-dependent set-up times, auxiliary resources and machine down time. Ann. Oper. Res. **180**(1), 165–196 (2010)