Mediterranean Smart Cities Conference (MSCC 2024)
2-4 May, 2024, Martil, Morocco

Fuzzy Takagi-Sugeno Simulated Annealing for simple assembly line balancing problem

Hibatallah Meliani
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
hibatallah_meliani@um5.ac.ma

Mohamed Hamza Boulaich
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
mohamedhamza_boulaich@um5.ac.ma

Said Ohamouddou
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
said_ohamouddou@um5.ac.ma

Hanaa El Afia
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
hanaa_elafia@um5.ac.ma

Abdellatif El Afia
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
abdellatif.elafia@ensias.um5.ac.ma

Abstract—**The assembly line process is one of the important processes in manufacturing industries. It performance has a big influence on performance of the global manufacturing process. Assembly Line Balancing (ALB) is the assignment of a set of tasks to workstations while respecting that precedence relations among tasks are satisfied and optimizing different objectives. In this paper, we propose a Fuzzy Takagi-Sugeno Simulated Annealing Algorithm (FTSSAA), that self-adapts it general parameters using a fuzzy model with a Takagi-Sugeno (TSK) controllers to solve the Simple Assembly Line Balancing Problem of type I (SALBP-I). The performance of the proposed model was verified against a standard Simulated Annealing algorithm on different instances of SALBP-I**

Keywords— Simulated Annealing, Fuzzy logic system, Adaptive parameter control, Takagi-Sugeno (TSK) controllers, Assembly Line Balancing Problem

I. INTRODUCTION

There are two groups in the Assembly Line Balancing Problem (ALBP), the simple ALBP (SALBP) and the generalized ALBP (GALBP). GALBP is the more complex version of SALBP where more real-world characteristics such as U-shaped line, mixed-model and stochastic task times are taken into consideration.

SALBP has been among the well-studied problems in the line balancing context and most of the research on ALBPs have been dealing with SALBP.

There are two types of SALBP. SALBP type1, where we try to minimize the number of workstations (m) given a cycle time (CT) and SALBP type2, where CT is minimized given m,

In this paper, we will focus on the Simple Assembly Line Balancing Problem type 1 (SALBP-1)

The major objective of SALBP-1 is to reduce the number of workstations:

$$\text{Minimize } \left(m \times \sqrt{\sum_{k=1}^{m} (C - t_k)^2}\right) \quad (1)$$

With respect to:

$$\sum_{k \in M} x_{ik} = 1, \forall i \quad (2)$$

$$\sum_{k \in M} k..x_{ik} \leq \sum_{k \in M} k..x_{jk}, \forall P^{'}(i,j) = 1 \quad (3)$$

$$\sum_{i \in N} x_{ik}.t_i \leq C, \forall k \quad (4)$$

$$x_{ik} \in \{0,1\}, \forall i,k \quad (5)$$

Where m is the number of stations, is the cycle time and t_k is the workload at station k.

The notations are described as follows:
C: the cycle time.
i: the task identifier, where $i \in N$
k: the workstation identifier, where $k \in M$
n: the number of tasks.
m: the number of workstations in the assembly line.
M: the set of workstations, where $M = \{1,2, . . , m\}$
N: the set of tasks.
P': the precedence matrix, $P^{'}(i,j)$ = 1, if task i is an immediate or indirect predecessor of task j; $P^{'}(i,j)$ = 0, otherwise.
(k): the set of tasks in workstation k
t_i : the processing time of the task i

x_{ik} : the binary variable. x_{ik} = 1, if task i is assigned to workstation k; x_{ik} = 0; otherwise.

Equation (2) make sure that every task is assigned to one workstation. Constraint (3) defines the precedence relationships between tasks i and j. Constraint (4) ensures that every workstation's processing time is not greater than the cycle time. Equation (5) means that xij variables are binary.

Among the approaches used to solve the ALBP, we find JAYA algorithm [1], Variable Neighborhood Search [3], Evolutionary Algorithm [4], Mathematical model and Dragonfly Algorithm [5]. In this paper, we are going to use a Fuzzy Takagi-Sugeno Simulated Annealing Algorithm (FTSSAA) that self-adapts it general parameters using a fuzzy model with a Takagi-Sugeno (TSK) controller. In the next section, we are going to present some algorithms used to solve ALBP. Section 3 presents the proposed model. In section 4, we are going to evaluate our model. Finally, we will conclude our work and propose future research related to our model.

II. RELATED WORK

In this work we are going to use an adaptive SA. Many improvements have been made to SA to solve ALBP Such as a Mathematical Model and a Simulated Annealing Algorithm in [12]. an adaptive simulated annealing (ASA) using modified COMSOAL algorithm for initial solution creation and two ASA algorithms for solution improvement in [13]. A genetic algorithm simulated annealing in [14].

Others improved SA by adapting the parameters using fuzzy logic system. In [15], Mamdani controllers were used to control the cooling speed of the simulated annealing algorithm. In [10], the same controllers adjust the neighborhood structure. In [16], fuzzy logic systems focus on the current temperature and the acceptance probability. In [17], a type2 fuzzy logic adjusts the same parameters.

fuzzy logic systems were also used in other metaheuristics such as Ant Colony System (ACS) to control the local pheromone [18], the pheromone parameters [2][19] and the population size [20],

To adapt the parameter of SA, other methods such as Hidden Markov Model (HMM) [21] were used [30] [31]. In [22] HMM adapts the cooling schedule of the SA temperature, the neighborhood structure in [23]. HMM was also able to adapt parameters of population based metaheuristics such as ACS [8] [24]-[27].

To our knowledge, no one has ever controlled all the parameters of the SA using fuzzy logic with Takagi-Sugeno (TSK) inference system. In this paper, we propose FTSSAA a self adaptive algorithm that uses fuzzy logic with Takagi-Sugeno (TSK) inference system to adjust the temperature, the acceptance probability and the local search operator of SA to solve SALBP type1.

III. PROPOSED APPROACH

A. *Simulated Annealing*

Simulated Annealing (SA) is a single solution based metaheuristics inspired by the annealing principles. proposed by Kirkpatrick et al. in 1983. SA finds a better solution by iteratively improving the initial solution.

The generalized Simulated Annealing (GSA) [6] is a general representation of the classical simulated annealing (CSA) and the fast simulated annealing (FSA). The temperature cooling schedule, the neighborhood structure and the acceptance probability are defined using mathematical equation.

In this work, we will use both the temperature and the acceptance probability functions of the GSA.

We replace qv notation in GSA original paper by qT. The temperature function is now defined as:

$$T_{qT}(t) = T_0 \frac{2^{qT-1}-1}{(1+t)^{qT-1}-1} \quad (6)$$

For the probability of acceptance, we choose to apply the generalized Metropolis algorithm used in [16] defined as:

$$p_{qa} = \min\left\{1, [1-(1-q_a)\Delta f\,]^{\frac{1}{1-qa}}\right\} \quad (7)$$

Where qa and qT are real numbers, Δf is the fitness difference and T_0 is the initial temperature.

The main idea of our work is to enhance the performance of GSA by controlling qa and qT and the local search operator choice using Fuzzy Logic system based on TSK inference system.

B. *Fuzzy Logic System*

Lotfi A Zadeh proposed the idea of Fuzzy logic in 1965. Fuzzy logic is a framework based on mathematics used for real world problems that deal with uncertainties. It involves soft or partial truth/false instead or binary crisp true or false.

Fuzzy logic involves key concepts such as fuzzy sets, which allow elements to have degrees of membership. Membership functions, defining the shape of these sets. Fuzzy rules expressing relationships between inputs and outputs. Fuzzification and defuzzification, converts crisp values to fuzzy sets and vice versa. The inference mechanism combines fuzzy rules, determining the degree to which each contributes to the output.

There are different types of fuzzy inference systems, in this paper we are going to focus on Takagi-Sugeno (TSK) controllers

TSK controllers [9] express rules as linear functions of the input variables. The output of a TSK controller is a weighted sum of the input variables, resulting in a non-fuzzy but crisp output.

C. SA CONTROL USING FUZZY LOGIC BASED ON TSK CONTROLLERS

The TSK inference system controls the qa and qT parameter using the current temperature and fitness difference as input (Figure2).
The rules defined for the temperature parameter qT and the acceptance parameter qa are represented in Figure 1

The output of TSK for qT is defined by the following equations:
For fast decrease:

$$z_1 = 0.0002 + 0.0006 \times x_1 + 0.0003 \times x_2$$

For decrease:

$$z_2 = 0.006 + 0.003 \times x_1 + 0.0007 \times x_2$$

For Slow Decrease:

$$z_3 = 0.009 + 0.0022 \times x_1 + 0.009 \times x_2$$

where x_1 and x_2 are the current temperature and the fitness difference of the SA respectively.
The coefficient of all equations used in TSK controller were found using a Genetic Algorithm.

		Fitness/Cost difference		
		Deteriorated	***No Change***	***Improved***
Temperature	High	Fast Decrease	Fast Decrease	Fast Decrease
	Medium	Decrease	Decrease	Slow Decrease
	Low	Decrease	Decrease	Slow Decrease

Figure 1. Rules of the proposed fuzzy system for parameter qT and qa

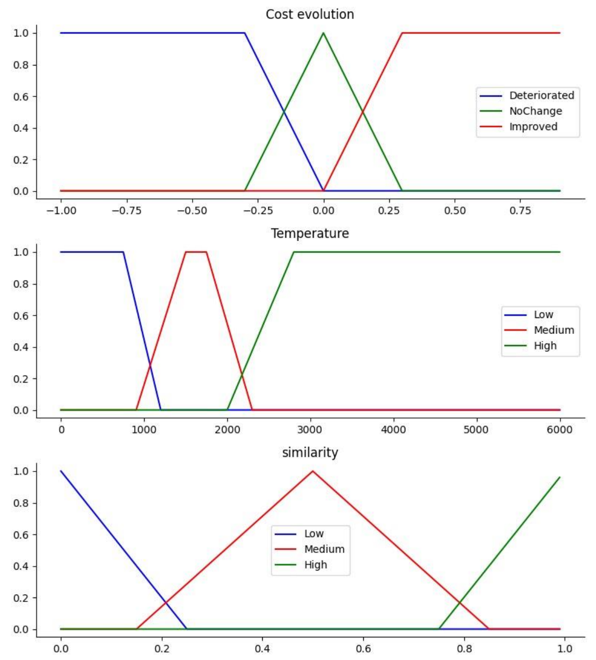

Figure 2. Antecedents Membership function Fuzzy Logic for the fitness difference, current temperature and similarity between solutions

To generate new solution, our model choses to apply a swap operator or generate completely a new individual based on the similarity between the current and the past solution, the current temperature and the fitness difference (Figure2).

The similarity between solution is calculated using ordered_jaccard_similarity () in Algorithm1. The ordered jaccard similarity is the size of the intersection of two individuals represented each by a list divided by the size of the union of the two list. The TSK then generate a destructive value between 0 and 1, if the value if greater than 0.5 the SA create a completely new solution else the algorithm applies a swap operator using generateSolution() function in Algorithm1.

The equations of the destruction value output has three inputs compared to qa and qT, which are the similarity between current and new solution, the fitness difference and the current temperature.

Therefore, our TSK controller generate three outputs: qa, qT and the destructive operator.

```
Algorithm1: The proposed FTSSAA algorithm

Input: Objective functions f, Maximum number of
iteration(MaxIter), initial temperature T_init

Output: best solution for the objective functions f
Initialize parameters T_init, qa (acceptance parameter), qt
(temperature parameter), op (operator_destruction_value)

Ti ←T_init
S ← initialSolution()
Sbest ← S

for  i ← 1 to MaxIter   do
     S' ← generateSolution(op, S)
     Δf ← f(S') – f(S)
     normΔf ← normalize(Δf )
     similarity ← ordered_jaccard_similarity (S', S)
     if Δf < 0 or (min {1, [1 - (1 - qa )Δf]^(1/(1-qa))} >

random[0,1]
               S ← S'
     end
     if f(S') < f(Sbest):
               Sbest ← S'
     end
     qt, qa, op←fuzzyLogicController(Ti, normΔf ,
similarity)
     Ti ← T_init (2^(qt−1) − 1) / ((1+i)^(qt−1) − 1)

end
```

IV. PERORMANCE EVALUATION

To study the behavior of our improved FTSSAA, we tested it on several SALBP benchmark instances from [2] using using Python language, SciKit-Fuzzy library and run on a PC with Intel Core i5 2.30GHz processor and 4 GB of RAM. The proposed algorithm is compared with

classical simulated annealing annealing (SA) with a cooling rate of 0.8 used in [28]. The maximum iteration is 100, the initial temperature is 5000. The algorithms were tested on every instance 30 times.
The results in table1 show that FTSSAA give better solution compared to SA.
The results signification is evaluated using the Wilcoxon signed-rank test with a significant level of 0.05. The p-values in table are less than 0.05 which mean that the proposed algorithm outperform the standard SA.

TABLE I. PERFORMANCE OF THE SA COMPARED TO FTSSAA ON ALBP INSTANCES

		FTSSAA	*SA*	*p-value*
n=20_455	Min	3	3	0.019
	Avg	**3.03**	3.26	
	CPU	0.29	0.15	
n=20_47	Min	4	4	0.0009
	Avg	**4.23**	4.6	
	CPU	0.24	0.12	
n=20_36	Min	14	14	0.0005
	Avg	**14.3**	15.83	
	CPU	0.27	0.14	
n=50_183	Min	34	34	0.018
	Avg	**35.9**	36.76	
	CPU	0.55	0.40	
n=50_408	Min	**30**	31	0.0003
	Avg	**32.26**	33.63	
	CPU	0.70	0.74	
n=50_494	Min	34	34	0.007
	Avg	**35.83**	36.9	
	CPU	0.78	0.64	
n=100_335	Min	**13**	14	0.157
	Avg	**13.96**	14.03	
	CPU	1.27	1.42	
n=100_445	Min	69	69	0.017
	Avg	**71.7**	72.93	
	CPU	2.23	1.56	
n=100_149	Min	68	68	0.008
	Avg	**71.13**	72.9	
	CPU	1.71	1.86	

V. CONCLUSION

In this paper, we tried to improve the performance of the SA in solving the Simple Assembly Line Problem (SALP). We compared the conventional simulated annealing and its fuzzy-controlled using TSK fuzzy inference system across different instances from benchmark datasets. The results show that our algorithm gives better solution compared to SA, which means that the fuzzy logic controller helps the SA search for better solution by controlling the local search operator type, the probability of the acceptance of the new generated solution and the cooling schedule. It shows the effectiveness of simulated annealing, with the fuzzy controller for the specified SALP instances. For future researches, we want to try complex ALP versions using the Fuzzy Takagi-Sugeno Simulated Annealing Algorithm (FTSSAA). We want also to extend the study to evaluate the performance of our algorithm and compare it to other advanced metaheuristic approaches.

.

REFERENCES

[1] Zhang, Z., Tang, Q., Han, D., & Qian, X. (2020). An enhanced multi-objective JAYA algorithm for U-shaped assembly line balancing considering preventive maintenance scenarios. International Journal of Production Research, 1–20. doi:10.1080/00207543.2020.1804639

[2] Bouzbita S., El Afia A., Faizi R., 2020, The behaviour of ACS-TSP algorithm when adapting both pheromone parameters using fuzzy logic controller, International Journal of Electrical and Computer Engineering (IJECE) Vol. 10, No. 5, October 2020, pp. 5436~5444

[3] Roshani, A., Paolucci, M., Giglio, D., & Tonelli, F. (2020). A hybrid adaptive variable neighbourhood search approach for multi-sided assembly line balancing problem to minimise the cycle time. International Journal of Production Research, 1–26. doi:10.1080/00207543.2020.1749958

[4] Qiuhua Tang, Kai Meng, Lixin Cheng, Zikai Zhang, An improved multi-objective multifactorial evolutionary algorithm for assembly line balancing problem considering regular production and preventive maintenance scenarios, Swarm and Evolutionary Computation,Volume 68,2022,101021,ISSN 2210-6502, doi: 10.1016/j.swevo.2021.101021.

[5] Beikun Zhang, Liyun Xu, Jian Zhang, Balancing and sequencing problem of mixed-model U-shaped robotic assembly line: Mathematical model and dragonfly algorithm based approach, Applied Soft Computing, Volume 98, 2021, 106739, ISSN 1568-4946,doi: 10.1016/j.asoc.2020.106739.

[6] Tsallis, C., & Stariolo, D. A. (1996). Generalized simulated annealing. Physica A: Statistical Mechanics and Its Applications, 233(1-2), 395–406.

[7] Agostini, F. P., Soares-Pinto, D. D. O., Moret, M. A., Osthoff, C., & Pascutti, P. G. (2006). Generalized simulated annealing applied to protein folding studies. Journal of Computational Chemistry, 27(11), 1142–1155.

[8] Bouzbita, S., El Afia, A., and Faizi, R. (2018, May). Parameter adaptation for ant colony system algorithm using Hidden Markov Model for TSP problems. In Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications (pp. 1-6). DOI: 10.1145/3230905.3230962

[9] TAKAGI, T., & SUGENO, M. (1993). Fuzzy Identification of Systems and Its Applications to Modeling and Control. Readings in Fuzzy Sets for Intelligent Systems, 387–403.

[10] Lalaoui, M., El Afia, A., & Chiheb, R. (2018). Simulated Annealing with Adaptive Neighborhood using Fuzzy Logic Controller. Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications - LOPAL '18

[11] Qin, L., Wang, J., Li, H., Sun, Y., & Li, S. (2017). An Approach to Improve the Performance of Simulated Annealing Algorithm Utilizing the Variable Universe Adaptive Fuzzy Logic System. IEEE Access, 5, 18155–18165.

[12] Yang, W., & Cheng, W. (2020). A Mathematical Model and a Simulated Annealing Algorithm for Balancing Multi-manned Assembly Line Problem with Sequence-Dependent Setup Time. Mathematical Problems in Engineering, 2020, 1–16. doi:10.1155/2020/8510253

[13] Güden, H., & Meral, S. (2015). An adaptive simulated annealing algorithm-based approach for assembly line balancing and a real-life case study. The International Journal of Advanced Manufacturing Technology. doi:10.1007/s00170-015-7802-y

[14] Zhong, Y. (2016). Hull mixed-model assembly line balancing using a multi-objective genetic algorithm simulated annealing optimization approach. Concurrent Engineering, 25(1), 30–40. doi:10.1177/1063293x16666204

[15] Lalaoui M., A. El Afia, R. Chiheb, (2017). Fuzzy logic controller for an adaptive Huang cooling of simulated annealing, The 2nd International Conference on Big Data, Cloud and Applications (CloudTech'17) IEEE Conference, Rabat, Morocco. Doi: 10.1145/3090354.3090420

[16] Lalaoui, M., & El Afia, A. E. (2018). A Fuzzy generalized simulated annealing for a simple assembly line balancing problem. IFAC-PapersOnLine, 51(32), 600–605.

[17] Lalaoui M., A. El Afia, (2018), A Versatile Generalized Simulated Annealing Using Type-2 Fuzzy Controller for the Mixed-Model Assembly Line Balancing Problem, the 9th IFAC Conference on Manufacturing Modelling, Management and Control, Berlin, Germany, August 28-30, 2019. Doi:10.1016/j.ifacol.2019.11.633.

[18] El Afia, A., Bouzbita, S., and Faizi, R. (2017). The effect of updating the local pheromone on acs performance using fuzzy logic. International Journal of Electrical and Computer Engineering, 7(4), 2161. DOI: 10.11591/ijece.v7i3

[19] Bouzbita, S., El Afia, A. (2022). A Fuzzy Meta Model for Adjusting Ant Colony System Parameters. In book: Proceedings of the 5th International Conference on Big Data and Internet of Things (pp.48-58). DOI:10.1007/978-3-031-07969-6_4

[20] Bouzbita, S., El Afia, A., and Faizi, R. (2019, September). Adjusting Population Size of Ant Colony System Using Fuzzy Logic Controller. In International Conference on Computational Collective Intelligence (pp. 309-320). Springer, Cham. DOI: 10.1007/978-3-030-28374-2_27

[21] Lalaoui M., A. El Afia, R. Chiheb.(2017). A Self-Tuned Simulated Annealing Algorithm Using Hidden Markov Model, International Journal of Electrical and Computer Engineering (IJECE). Doi:10.11591/ijece.v8i1.pp291-298

[22] Lalaoui M., A. El Afia and R. Chiheb, (2016). Hidden Markov Model for a Self-Learning of Simulated Annealing Cooling Law, the 5th International Conference on Multimedia Computing and Systems IEEE Conference, ICMCS'16, Marrakech, Morocco. Doi: 10.1109/ICMCS.2016.7905557

[23] Lalaoui M., El Afia A., Chiheb R. (2021). Dynamic Simulated Annealing with Adaptive Neighborhood Using Hidden Markov Model. In: Yalaoui F., Amodeo L., Talbi EG. (eds) Heuristics for Optimization and Learning. Studies in Computational Intelligence, vol 906. Springer, Cham. Doi:10.1007/978-3-030-58930-1_11.

[24] Bouzbita, S., El Afia, A., Faizi, R., and Zbakh, M. (2016, May). Dynamic adaptation of the ACS-TSP local pheromone decay parameter based on the Hidden Markov Model. In 2016 2nd international conference on cloud computing technologies and applications (CloudTech) (pp. 344-349). IEEE. DOI: 10.1109/CloudTech.2016.7847719

[25] Bouzbita, S., El Afia, A., and Faizi, R. (2016). A novel based Hidden Markov Model approach for controlling the ACS-TSP evaporation parameter. In 2016 5th international conference on multimedia computing and systems (ICMCS) (pp. 633-638). IEEE. DOI: 10.1109/ICMCS.2016.7905544

[26] Bouzbita, S., El Afia, A., and Faizi, R. (2019). Hidden markov model classifier for the adaptive ACS-TSP pheromone parameters. In bioinspired Heuristics for Optimization (pp. 153-169). Springer, Cham. DOI: 10.1007/978-3-319-95104-1_10

[27] Bouzbita, S., El Afia, A., and Faizi, R. A New Hidden Markov Model Approach for Pheromone Level Exponent Adaptation in Ant Colony System. In Heuristics for Optimization and Learning (pp. 253-267). Springer, Cham. DOI: 10.1007/978-3-030-58930-1_17

[28] Erel, E., Sabuncuoglu, I., & Aksu, B. A. (2001). Balancing of U-type assembly systems using simulated annealing. International Journal of Production Research, 39(13), 3003–3015. doi:10.1080/00207540110051905

[29] https://assembly-line-balancing.de/salbp/benchmark-data-sets-2013/

[30] Afia, A. E., Lalaoui, M., & Chiheb, R. (2019). A Self Controlled Simulated Annealing Algorithm using Hidden Markov Model State Classification. Procedia Computer Science, 148, 512–521. doi:10.1016/j.procs.2019.01.024

[31] Lalaoui, M., Afia, A. E., & Chiheb, R. (2017). A self-adaptive very fast simulated annealing based on Hidden Markov model. 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech). doi:10.1109/cloudtech.2017.8284698