

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Ant Colony Optimization for Multiobjective Buffers Sizing Problems

Hicham Chehade, Lionel Amodeo and Farouk Yalaoui  
*Université de Technologie de Troyes, Institut Charles Delaunay,  
Laboratoire d'Optimisation des Systèmes Industriels  
France*

## 1. Introduction

The optimization of production lines relies on many parameters such as the equipment selection, the buffers sizing, the line balancing or others. This is done in order to get high performed lines with the lowest costs so that line manufacturers remain competitive in the markets.

In this work, we are interested in the buffers sizing problem on assembly lines (figure 1). It is widely studied in the literature. However, few works are focused on solving multiobjective buffers sizing problems. Indeed, despite the large number of papers in the literature dealing with the buffers sizing problem, one can feel the lack in the works dealing with the multiobjective problems. In fact, we may notice that almost the total number of papers dealing with the buffers sizing problem takes in consideration one criterion at once (Altıparmak et al., 2002), (D’Souza & Khator, 1997), (Hamada et al., 2006). Some other works take in consideration more than one criterion but they use weighted sums in the fitness function (Abdul-Kader, 2006), (Dolgui et al., 2002). In 2007 and later in 2009, Chehade et al. have applied the multiobjective optimization using multiobjective ant colony optimization algorithms (Chehade et al., 2007), (Chehade et al., 2009).

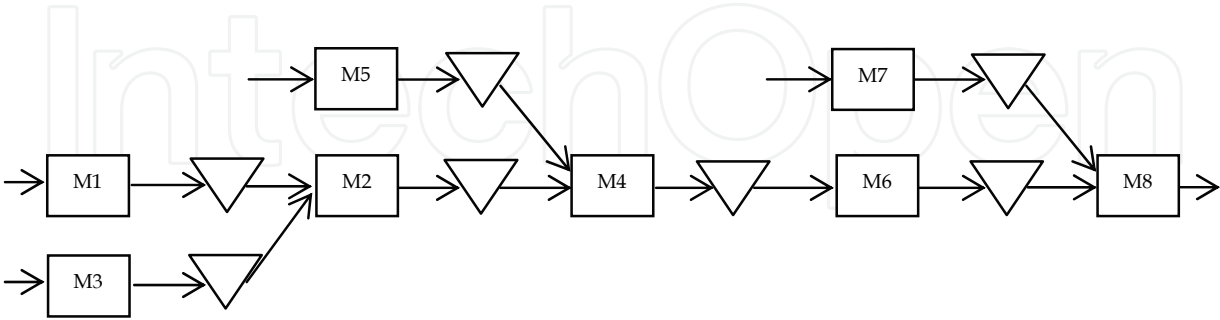


Fig. 1. An example of an assembly line with 8 machines (Mi) and 7 intermediate buffers

The lack of studies dealing with multiobjective buffers sizing problems is probably due to the complexity of the problem (Harris & Powell, 1999) and the difficulty to develop the suitable tool to optimize it efficiently. However, dealing with a single objective may lead to sidestep the problem (Collette & Siarry, 2002). Moreover, a multiobjective optimization

allows some freedom degrees that do not appear in the single objective optimization. Based on that, we have decided to continue studying this problem with its multiobjective type. The performances evaluations of the different configurations are done using the ARENA simulation software. Indeed, analytical methods seem to be inappropriate to solve hard and complex computational problems (Han & Park, 2002).

In this work, and similarly to our previous works, the problem consists of sizing the buffers of different stations in an assembly line taking in consideration that the size of each buffer is bounded by a lower and an upper value. Two objectives are taken in consideration: the maximization of the throughput rate and the minimization of the buffers total size. Our contribution in this work is to present two new multiobjective resolution methods for the studied problem. These two methods are first a multiobjective ant colony algorithm with a guided local search and then a Lorenz multiobjective ant colony algorithm.

In the literature and among the methods applied to solve the buffers sizing problems, the most efficient ones are metaheuristics for their ability to solve complex operations research problems. Indeed, the stochastic nature of these methods allows affording the combinatorial explosion in the number of possible solutions (Dréo et al., 2003). Genetic algorithms belong for example to metaheuristics which can be applied to solve buffers sizing problems (Dolgui et al., 2002), (Hamada et al., 2006). Simulated Annealing is another method applied in many works (Spinellis et al., 2000), (Papadopoulos & Vidalis, 2001). The literature presents also other methods such as neural networks (Altıparmak et al., 2002) or Tabu search (Lutz et al., 1998).

In this work, the different resolution methods are therefore based on ant colony optimization. The first one, initially presented in a previous work (Chehade et al., 2009) is a multiobjective ant colony optimization algorithm (MOACS). Ant colony optimization algorithms have been, until a recent period, applied on single objective problems. Initially developed to solve the travelling salesman problem (Dorigo & Gambardella, 1997), they have been later applied efficiently in different other fields such as scheduling or line balancing. Recently, we have noticed the development of the first algorithms based on ant colony optimization to solve multiobjective optimization problems (Benlian & Zhiquan, 2007), (Pellegrini et al., 2007).

Afterthat, a guided local search is coupled to the MOACS algorithm (MOACS-GLS) in order to enhance the performances of the latter. Indeed, the guided local search is used to avoid the occurrence of local optimum solutions. Voudouris and Tsang (Voudouris & Tsang, 1996) were the first to introduce a general optimization technique suitable for a wide range of combinatorial optimization problems which is the guided local search. This metaheuristic is used to guide the search out of a local optimum and it was successfully applied to many practical problems such as frequency applications (Voudouris & Tsang, 1996), vehicle routing problems (Kilby et al., 1999) and the quadratic assignment problem (Hani et al., 2007).

The last algorithm is a multiobjective ant colony algorithm but using the Lorenz dominance (L-MOACS). The Lorenz dominance relationship, as shown below, allows retaining the solutions which better fit the objectives of the problem. This is done by providing a better domination area by rejecting the solutions founded on the extreme sides of the Pareto front. The Lorenz dominance relationship also called equitable dominance relationship was defined in 1999 (Kostreva & Ogryczak, 1999) and extended in 2004 (Kostreva et al., 2004). This relationship, which is considered as equitable and rational, has been applied efficiently in many works (Perny et al., 2004), (Dugardin et al., 2009a), (Dugardin et al., 2009b).

The remainder of this work is organized as follows. The problem description is in section 2. Section 3 presents the resolution methods. Computational experiments are presented in section 4 and we finish by a conclusion and perspectives in section 5.

## 2. Problem description

In this work, we are interested in sizing the buffers of assembly lines. The structures of the lines are formed by unreliable machines with exponential distribution and finite buffers. The lines have  $N$  machines and  $N-1$  intermediate buffers taking in consideration that each two workstations are separated by one buffer. The goal of our study is therefore to identify the best size for each intermediate buffer taking in consideration the characteristics of the machines. We assume here that the size of each buffer is bounded by a lower ( $l$ ) and an upper value ( $u$ ).

The objectives of this problem are as follows. For the first objective, we aim to maximize the throughput rate of the lines. As for the second objective, we look at minimizing the total size of the buffers. Therefore, the optimal configurations of the multiobjective problem are the lines configurations that give the highest throughput rates with the lowest buffers sizes.

The mathematical formulation of this multiobjective buffers sizing problem can be stated as follows. The objective function  $Z$  to be optimized (see (1)) depends on the two studied objectives: the maximization of the throughput rate  $E$  (see (2)) and the minimization of the total size of the buffers in the line (see (3)). The decision variable is  $Y_{ij}$  (size  $i$  of buffer  $j$ ).

$$Z = (O1, O2) \quad (1)$$

$$O1 : \text{Maximize}(E) \quad (2)$$

$$O2 : \text{Minimize} \left( \sum_{j=1}^{N-1} \sum_{i=1}^B Y_{ij} \cdot b_{ij} \right) \quad (3)$$

Subject to:

$$E = f(Y_{ij}) \quad (4)$$

$$E_{lj}^B \leq E_{ij}^B \leq E_{uj}^B; \forall i = 1, \dots, B; \forall j = 1, \dots, N-1 \quad (5)$$

$$\sum_{j=1}^{N-1} \sum_{i=1}^B Y_{ij} \cdot b_{ij} \leq S_{\max} \quad (6)$$

$$\sum_{i=1}^B Y_{ij} = 1; \forall j = 1, \dots, N-1 \quad (7)$$

$$Y_{ij} \in \{0, 1\}; \forall i = 1, \dots, B; \forall j = 1, \dots, N-1 \quad (8)$$

Notation:

$N$ : the number of machines in the line.

$N-1$ : the number of intermediate buffers in the line.  
 $B$ : the number of the possible sizes of a buffer.  
 $S_{max}$ : the total available space for the buffers of the line.  
 $Y_{ij}$ : a binary variable equals 1 if buffer size  $i$  is assigned to buffer  $j$  and 0 otherwise.  
 $b_{ij}$ : the size  $i$  of buffer  $j$ .  
 $E$ : the throughput rate of the line  
 $E_{ij}^B$ : the throughput rate of a buffer  $j$  while having a size  $i$ .  
 $E_{lj}^B$ : the lower bound for the throughput rate of a buffer  $j$ .  
 $E_{uj}^B$ : the upper bound for the throughput rate of a buffer  $j$ .

Constraint (4) indicates that the throughput rate  $E$  of the line is a function of the sizes of the buffers. Constraint (5) shows that the size of each buffer is bounded by a lower ( $l$ ) and an upper value ( $u$ ). Constraint (6) means that the total space of the buffers must not exceed the total available space for the buffers of the line. Constraint (7) imposes that a unique size must be assigned to each buffer. Constraint (8) defines the binary decision variables.

For the performances evaluations of the different tested configurations, we have used the discrete event simulation through the ARENA simulation software.

3. Resolution methods

In this section, we discuss the resolution methods that we have developed for our multiobjective problem. First, we present the multiobjective ant colony optimization algorithm (MOACS) and then we present the Lorenz dominance relationship. Finally, we present the overall algorithm that we call L-MOACS (Lorenz Multiobjective Ant Colony Algorithm).

3.1 Multiobjective Ant Colony Algorithm

In this section, we present the multiobjective ant colony system algorithm (MOACS) that we have applied in this work. It is based on the general structure of a classical ant colony optimization algorithm by being based on four main steps: the solutions encoding, the ants tours construction, the local pheromone updates and the global pheromone updates. The only difference is related to the number of the pheromone matrices. In fact, knowing that the multi objective optimization takes in consideration different criterion simultaneously, we consider that the total number of pheromone matrices should be equal to the number of objectives to be optimized. Therefore, we have in this work two pheromone matrices.

Solutions encoding

This encoding is presented in figure 2 where we consider that we have  $N$  stations and  $N-1$  intermediate buffers. A lower ( $l_j$ ) and an upper ( $u_j$ ) value bound the capacity of each buffer  $j$ .

Buffer 1	$l_1$	$l_1 + 1$	...	$u_1 - 1$	$u_1$
Buffer 2	$l_2$	$l_2 + 1$	...	$u_2 - 1$	$u_2$
:			:		
Buffer $N-1$	$l_n$	$l_n + 1$	...	$u_n - 1$	$u_n$

Fig. 2. Solutions encoding

### Tours construction

First, each ant is deposited randomly on a starting point which represents the size that has to be assigned to the first buffer. After that, an ant  $k$  chooses to move from a point  $r$  to another point  $s$  based on (9).

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \left\{ \left[ \sum_{o=1}^O w_o \tau_{r,u}^o \right]^\alpha \eta_{r,u}^\beta \right\} & \text{if } q \leq q_0 \\ S^* & \text{otherwise} \end{cases} \quad (9)$$

Where:

$O$  : the number of the considered objectives.

$\tau_{r,s}^o$  : the quantity of pheromone between the points  $r$  and  $s$  based on objective  $o$ .

$w_o$  : the coefficient of importance granted to each objective (we consider that  $w_1 = w_2 = 0.5$ ).

$q$  : is a random number generated between 0 and 1.

$q_0$  : is a parameter ( $0 \leq q_0 \leq 1$ ) which determines the relative importance of exploitation against exploration.

$S^*$  : is a random variable chosen based on a probability given by (10).

$\eta_{r,s}$  : is a static value used as a heuristic of innate desirability to choose  $s$  starting from  $r$  and is also called the ant visibility to choose a point starting from another point.

$\alpha$  and  $\beta$  are used to determine the relative importance of pheromones versus the visibility.

$J_k(r)$  : is the set of points not yet visited by ant  $k$ .

$$S^* = \begin{cases} \frac{\left[ \sum_{o=1}^O w_o \tau_{r,s}^o \right]^\alpha [\eta_{r,s}]^\beta}{\sum_{u \in J_k(r)} \left[ \sum_{o=1}^O w_o \tau_{r,u}^o \right]^\alpha [\eta_{r,u}]^\beta} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

### Local pheromone updates

The local pheromone update is applied once all ants have finished their tours. It is computed based on (11) where  $\tau_0$  is the initial quantity of pheromone.

$$\tau_{r,s}^o = (1 - \rho) \cdot \tau_{r,s}^o + \rho \cdot \tau_0 \quad (11)$$

### Global pheromone updates

The global pheromone update is realized according to (12) while being based on the non dominated solutions obtained at each generation.

$$\tau_{r,s}^o = (1 - \rho) \cdot \tau_{r,s}^o + \rho \cdot \Delta \tau_{r,s}^o \quad (12)$$

$\Delta \tau_{r,s}^o$  consists on supporting the non dominated solutions found so far. It is computed based on (13) and (14) where  $C_{gb}$  and  $E_{gb}$  represent, respectively, the lowest total size of buffers and the highest throughput rate found so far by ants.



$$\Delta\tau_{r,s}^1 = \begin{cases} (C_{gb})^{-1} & \text{if } r,s \in \text{non-dominated-solutions} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$\Delta\tau_{r,s}^2 = \begin{cases} E_{gb} & \text{if } r,s \in \text{non-dominated-solutions} \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

### 3.2 Guided local search

In order to adapt the guided local search and to couple it with the MOACS algorithm, a local search procedure based on the neighborhood search is first applied. This local search is applied on the first optimal front obtained after the ants tours and before discarding the non feasible solutions. Afterthat, we will now be able to check if the space constraint is satisfied or not.

In fact, if the total space of the buffers according to a given solution in the optimal front is higher than the allowed one ( $S_{max}$ ), then we try to minimize it by decreasing the capacity of each buffer by one. This is done on all the buffers. Note that as we are limited by a lower and an upper bound for buffers, we are not able, all the times, to apply this procedure. For example, we may not be able to decrease the size of a given buffer (if its size is equal to the lower value  $l$ ).

In the other side, if the total covered space is smaller than the allowed one, we try to look for a neighborhood solution which can decrease the total size of the buffers and increase the throughput rate. We increase then at each time the size of the buffers by one.

At the end of the local search, we take in consideration the non dominated solutions according to the Pareto dominance rule. As using only the local search may lead to local optimum cases and not necessarily to global ones, we have seen that the best way to overcome this lack is by applying the guided local search metaheuristic as explained below.

The purpose of making a local search procedure getting out of a local optimum is satisfied by adapting the guided local search metaheuristic to our problem. That is based on using augmented objective functions by adding penalties to the initial objective functions. Therefore, we associate features to the objective functions taking in consideration that those features are in the local optimum. The features are selected based on the type of the problem to be optimized. However, each feature  $f_i$  must have the following components:

- An indicator function  $I_i(s)$  indicating whether the feature is present in the current solution  $s$  ( $I_i(s) = 1$ ) or not ( $I_i(s) = 0$ ).
- A cost function  $c_i(s)$  giving the cost of having  $f_i$  in  $s$ .
- A penalty factor  $p_i$  (initially set to 0) to penalize the occurrence of  $f_i$  in local optimum.

We are using here one feature ( $f_1$ ) in our case as we have one element which is the buffers sizing. Based on the local search procedure, the indicator function  $I_1$  of feature  $f_1$  is equal to 1 if we are able to modify the size of a given buffer and 0 otherwise (if the size of the buffer is equal to the lower  $l$  or the upper  $u$  value). The occurrence of a local optimum allows the manipulation of the augmented cost function by applying a penalty modification procedure. The penalty parameters  $p_i$  are incremented by one for all features  $f_i$  that maximize an expression called utility as shown in (15) where  $c_1(s)$  is computed according to (16) and  $c_2(s)$  according to (17).

$$util(s, f_1) = I_1(s) \cdot c_1(s) / (1 + p_1) \quad (15)$$

$$c_1(s) = \sum_{j=1}^{N-1} \sum_{i=1}^B Y_{ij} \cdot b_{ij} \quad (16)$$

$$c_2(s) = 1 / E \quad (17)$$

The augmented cost function is present in (18) for the space objective and in equation (19) for the throughput rate objective. It is used to make the local optimum more costly than other solutions in the surrounding search space in order to avoid it.

$$h_1(s) = c_1(s) + \lambda_1 \cdot I_1(s) \cdot p_1 \quad (18)$$

$$h_2(s) = c_2(s) + \lambda_2 \cdot I_1(s) \cdot p_1 \quad (19)$$

Where  $\lambda$  is a parameter for controlling the constraints strength with respect to the solution objective.  $\lambda_1$  is computed based on the total space of the buffers of the line and the different possibilities of the sizes of the buffers as shown in (20) where  $B$  is the number of possible sizes for each buffer.  $\lambda_2$  is computed as illustrated in (21).

$$\lambda_1 = \left( \sum_{j=1}^{N-1} \sum_{i=1}^B Y_{ij} \cdot b_{ij} \right) / (2 \cdot B) \quad (20)$$

$$\lambda_2 = (1 / E) / (2 \cdot B) \quad (21)$$

To resume the adaptation of the guided local search metaheuristic to our problem, starting from the set of the non dominated solutions obtained by the MOACS algorithm, a local search procedure is applied to find local minima with respect to the augmented cost function. If those minima have objective functions (not augmented) better (smaller for the space objective and greater for the throughput objective) than the best objective functions ever found, they are saved as non dominated solutions. Finally, the configurations having the maximum utilities would have their penalties increased. The process is repeated until a stopping criterion. At this step, we are able to identify the feasible non dominated solutions.

### 3.3 Multiobjective Ant Colony algorithm with a Guided Local Search

The structure of the multiobjective ant colony algorithm with the guided local search (MOACS-GLS) is presented by algorithm 1.

Algorithm 1: MOACS-GLS algorithm

Step 1: Parameters initialization

Step 2: For the  $k$  ants

Compute the desirability factors associated with each objective, so as to select the successive nodes according to the visibility factor and the pheromone trails

Apply a local update of the pheromone trails

Apply the guided local search procedures

Apply a global update of the pheromone trails while being based on the non dominated solutions

End For

Step 3: Iterate from Step 2 until a stopping criterion



3.4 The Lorenz dominance relationship

The Lorenz dominance relationship restricts the Lorenz searching space to a subset of the Pareto searching space. Thus, the Lorenz dominance increases the speed of the algorithm by focusing on the promising solution set 0. It provides then a larger domination area to each solution which implies the rejection of the solutions that are at the furthest end of a Pareto front.

Indeed, let us take in consideration, for example, two criterion  $f_1$  and  $f_2$  to be minimized. Figure 2 describes the Pareto dominance area of a solution X. With the Pareto dominance, the solution X dominates any other solution  $X'$  that has either  $f_1(X')$  or  $f_2(X')$  higher than  $f_1(X)$  or  $f_2(X)$ . However, for the same point X, the Lorenz dominance area is shown in figure 3. It is clear that the Lorenz dominance area is larger than the Pareto dominance area.

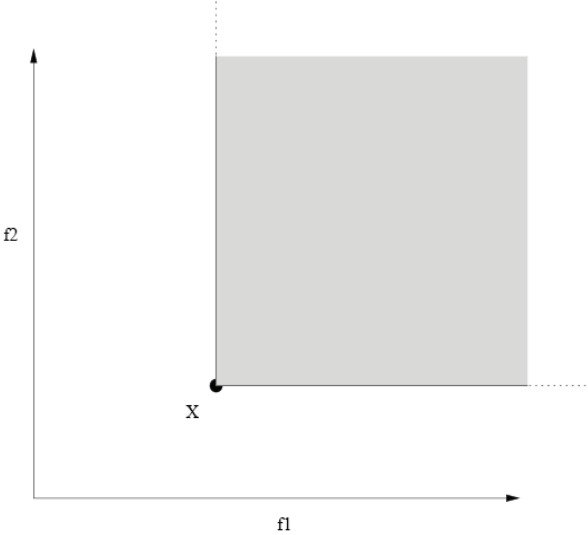


Fig. 2. Pareto dominance area (Dugardin et al. 2009a)

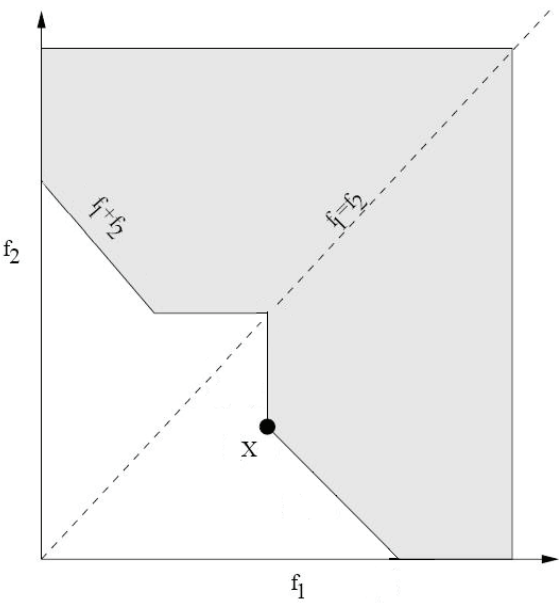


Fig. 3. Lorenz dominance area (Dugardin et al. 2009a)

Using the Lorenz dominance, denoted by  $\prec_L$ , we can consider that  $X$  is Lorenz-optimal if, and only if, no solution  $X'$  exists so that  $f(X') \prec_L f(X)$ . Given a vector  $y(y_1, y_2)$  with two components, the corresponding Lorenz generalized vector is  $y_L(\max(y_1, y_2), y_1+y_2)$ . For more details about the Lorenz dominance relationship and its different characteristics and principles, reader is referred to the work of Dugardin et al. (Dugardin et al., 2009a) where a deep explanation about the Lorenz dominance is presented.

### 3.5 Lorenz Multiobjective Ant Colony algorithm

As mentioned before, the Lorenz Multiobjective Ant Colony algorithm (L-MOACS) is based on a multiobjective ant colony system algorithm using the Lorenz dominance relationship. Therefore, once all ants have constructed their tours and that local pheromone updates are applied, we identify the non dominated solutions based on the Lorenz dominance. The set of the Lorenz non dominated solutions, on which we should be based to globally update the pheromone matrices, constitute the optimal front of the problem. The overall L-MOACS algorithm is shown in algorithm 2.

Algorithm 2: L-MOACS algorithm

*Step 1: Initialization of parameters*

*Step 2: For all ants:*

- Assign sizes to buffers
- Local pheromone updates
- Identification of the non dominated solutions based on the Lorenz dominance
- Global pheromone updates

*Step 3: Return to step 2 until a stopping criterion is satisfied*

### 3.6 Comparison criteria

In this section, we describe the methods used to compare two optimal fronts ( $F1$  and  $F2$ ) obtained by the different algorithms (MOACS, MOACS-GLS and L-MOACS) in order to get an idea and to evaluate the performances of the solutions of each algorithm. Three comparison criteria are used: the number of solutions  $N_i$  in an optimal front  $i$ , the distance proposed by Riise  $\mu$  (Riise, 2002), the Zitzler measure  $C_i$  (Zitzler & Thiele, 1999).

The distance of Riise,  $\mu$ , is computed as the sum of the distances  $d_x$  between a solution  $x$  belonging to front  $F1$  and its orthogonal projection on front  $F2$  ( $\mu = \sum_{x=1}^N d_x$ ).  $\mu$  is negative if  $F1$

is under  $F2$  and positive otherwise. As the value of  $\mu$  depends on the number of solutions  $N_i$  in each front, a normalized value ( $\mu^* = \mu / N_i$ ) is generally taken in consideration.

The Zitzler measure  $C_1$  represents the percentage of solutions in  $F1$  dominated by at least one solution in  $F2$ . Taking in consideration that the measure is not symmetric, it is advised to compute  $C_2$  as well. In conclusion, a front  $F1$  is better than another front  $F2$  if  $C_1$  is smaller than  $C_2$ .

## 4. Computational experiments

Computational experiments are realized on assembly lines with finite buffers and where the machines time between failures, the times to repair and the processing times are exponentially distributed.

In order to compare the three algorithms, MOACS, MOACS-GLS and L-MOACS, several tests have been first realized so that we can set efficiently the different parameters of the algorithms. Final values are determined as a compromise between the quality of the final solutions and the convergence time needed. Those final values are as follows:  $k = 20$ ,  $\rho = 0.7$ ,  $q_0 = 0.7$ ,  $a = 0.7$ ,  $\beta = 0.3$ ,  $\tau_0 = 2$ . The stopping criterion for the ant colony algorithms is a given number of generations and is fixed at 100. The stopping criterion for the guided local search is a number of iterations which is fixed at 50.

Once the parameters values have been set, we apply the three algorithms on different instances. Two assembly lines are tested. The first is a non balanced assembly line where the machines have different processing times ( $T_i$ ). This is the first problem P1. The second concerns a balanced assembly line where the different machines have the same processing time ( $T_i$ ). This is then the second problem P2.

For the input data of the machines characteristics, we have adopted those used by Nahas et al. (Nahas et al., 2006). Those characteristics are the processing times ( $T_i$ ), the mean times to repair ( $MTTR$ ) and the mean times to failure ( $MTTF$ ). The values of these parameters are presented in tables 1 and 2.

Machine	MTTR	MTTF	$T_i$
1	450	820	40
2	760	5700	34
3	460	870	39
4	270	830	38
5	270	970	37
6	650	1900	40
7	320	1100	43
8	480	910	39
9	340	1050	41

Table 1. Input data for problem P1 (Nahas et al., 2006)

Machine	MTTR	MTTF	$T_i$
1	7	20	1
2	7	30	1
3	5	22	1
4	10	22	1
5	9	25	1
6	14	40	1
7	5	23	1
8	8	30	1
9	10	45	1

Table 2. Input data for problem P2 (Nahas et al., 2006)

Each buffer may have a size that is bounded between the lower value ( $l$ ) which is equal to 1 and the upper value ( $u$ ) which is equal to 20. Three different assembly line structures are tested for the two problems. Each structure is different from the other one by the number of machines  $N$  and thus the number of buffers  $N-1$ .  $N$  is equal to 3, 7 and 9 for the first (S1), second (S2) and third (S3) structures respectively.

We have realized three comparisons. The first one is between the MOACS-GLS and the MOACS algorithms and is presented in table 3. Table 4 shows the second comparison which is between the L-MOACS and the MOACS algorithms. The last comparison is between the L-MOACS and the MOACS-GLS algorithms and it is presented in table 5.

The results of the generated instances for the comparison between the MOACS-GLS and the MOACS algorithms are compared in table 3. It shows a comparison between the best fronts with the non dominated solutions obtained for each algorithm.  $N_1$  stands for the number of solutions in the optimal front obtained by the MOACS-GLS algorithm and  $N_2$  for the MOACS algorithm. The distance of Riise is presented under the  $\mu^*$  column.  $C_1$  and  $C_2$  stand for the Zitzler measure.  $C_1$  is a measure for front  $F1$  of the MOACS-GLS algorithm and  $C_2$  for that of front  $F2$  of the MOACS algorithm.

The comparisons are done on all of the 6 different tests realized for the two problems. To better explain table 3 for the reader, let us take the first line as an example. It shows a comparison between the two algorithms (MOACS-GLS and MOACS) applied on problem P1 and on an assembly line with 3 machines and 2 buffers (structure S1). The number of non dominated solutions ( $N_1$ ) in the MOACS-GLS best front is 15 and is equal to that of the MOACS Pareto front. The negative value of the  $\mu^*$  column (-0.01) shows that  $F1$  is under  $F2$ . Columns 6 and 7 show that 6.67% of solutions in the MOACS-GLS best front are dominated by at least one solution in the MOACS front while 13.33% of solutions in the MOACS front are dominated by at least one solution in the MOACS-GLS front. The same logic is followed to read the comparison parameters in the rest of the table.

		$N_1$	$N_2$	$\mu^*$	$C_1$	$C_2$
P1	S1	15	15	-0.01	6.67	13.33
	S2	16	16	-0.02	6.25	18.75
	S3	17	16	-0.03	5.88	37.50
P2	S1	16	16	-0.04	6.25	43.75
	S2	19	18	-0.05	0.00	44.44
	S3	19	18	-0.06	0.00	50.00
Mean		17	16.5	-0.03	4.175	34.62

Table 3. Comparison between MOACS-GLS (F1) and MOACS (F2) algorithms

Based on all the tested configurations, we may conclude that on the majority of the generated instances, the MOACS-GLS performs more efficiently than the MOACS algorithm. Indeed, we have noticed that as the size of the problem increases, the advantages of the MOACS-GLS algorithm get more obvious. In general, the mean value of the number of non dominated solutions for the MOACS-GLS is 17 against 16.5 for the MOACS. The mean value of the Riise distances is -0.03 which means that the optimal front of the MOACS-GLS ( $F1$ ) is under that of the MOACS algorithm ( $F2$ ). That means that we are maximizing the throughput rate of the line and at the same time minimizing the total size of the buffers when we apply the MOACS-GLS algorithm compared to the application of the MOACS algorithm. Finally, there is a general mean of 4.175% of solutions in the  $F1$  fronts that are dominated by at least one solution from the  $F2$  fronts against 34.62% of the  $F2$  solutions that are dominated by at least one solution from the  $F1$  fronts.

Once the first comparison is done, we have realized the second one between the L-MOACS algorithm based on the Lorenz dominance relationship and the MOACS algorithm using the Pareto dominance relationship. Table 4 shows the comparison between fronts  $F1$  (L-MOACS) and  $F2$  (MOACS). The same interpretation used to analyze table 3 is applied for table 4 with the three comparison criteria:  $N_1$  and  $N_2$ ,  $\mu^*$ ,  $C_1$  and  $C_2$ .

The first criterion which is the number of non dominated solutions shows that there are less non dominated solutions in the L-MOACS optimal front ( $N_1$ ) than in the MOACS optimal front ( $N_2$ ). On all the tested structures, the mean number of solutions with the Lorenz dominance is equal to 13.5 against 16.5 with the Pareto dominance. This is due to the fact that, with the Lorenz dominance, we can get fewer solutions compared to the Pareto dominance. The second criterion ( $\mu^*$ ) shows the clear advantage of the L-MOACS algorithm compared to the MOACS algorithm. The value of  $\mu^*$  is always negative for all the tested instances which means that the optimal front of the L-MOACS algorithm fits better the two objectives of the problem (maximization of the throughput rate and minimization of the total size of the buffers). The same conclusion may be deduced based on the third criterion when one can notice that the mean percentage of solutions in the L-MOACS front that are dominated by at least one solution in the MOACS front (3.86%) is largely smaller than that of the MOACS front (43.84%).

		$N_1$	$N_2$	$\mu^*$	$C_1$	$C_2$
P1	S1	13	15	-0.03	7.69	20.00
	S2	12	16	-0.03	8.33	31.25
	S3	13	16	-0.05	0.00	56.25
P2	S1	14	16	-0.04	7.14	50.00
	S2	14	18	-0.07	0.00	44.44
	S3	15	18	-0.08	0.00	61.11
Mean		13.5	16.5	-0.05	3.86	43.84

Table 4. Comparison between L-MOACS (F1) and MOACS (F2) algorithms

The third comparison is between the L-MOACS algorithm and the MOACS-GLS algorithms. Table 4 presents first the number of non dominated solutions  $N_1$  and  $N_2$  in fronts  $F1$  (for the L-MOACS) and  $F2$  (for the MOACS-GLS) respectively. As in the comparison between L-MOACS and MOACS, the number of non dominated solutions with the Lorenz dominance relationship is fewer than that with the Pareto dominance relationship. For the two other criteria, we may notice a small advantage for the L-MOACS algorithm compared to the MOACS-GLS algorithm. Indeed, taking in consideration the Riise distance, the two algorithms have the same performances for 2 instances over 6 tested ones. In general, the mean value of  $\mu^*$  is only equal to -0.01. That small advantage may be confirmed while being based on the Zitzler measure. Here also, for two tested instances the two algorithms have the same performances. For the four other instances, the mean percentage of L-MOACS solutions that are dominated by at least one MOACS-GLS solution ( $C_1$ ) is equal to 2.67% against 5.63% for  $C_2$ .

		$N_1$	$N_2$	$\mu^*$	$C_1$	$C_2$
P1	S1	13	15	-0.00	0.00	0.00
	S2	12	16	+0.01	8.33	6.25
	S3	13	17	-0.02	7.69	11.76
P2	S1	14	16	-0.00	0.00	0.00
	S2	14	19	-0.01	0.00	0.00
	S3	15	19	-0.03	0.00	15.79
Mean		13.5	17	-0.01	2.67	5.63

Table 5. Comparison between L-MOACS (F1) and MOACS-GLS (F2) algorithms



In conclusion, and taking in consideration the three comparisons between the three applied algorithms, we may deduce that while applying the Pareto dominance relationship, hybrid algorithms perform better than classic ones. That was obvious while comparing the classical multiobjective ant colony algorithm (MOACS) and the hybrid algorithm which is a multiobjective ant colony algorithm with a guided local search (MOACS-GLS). In addition to that, we may notice the advantages of the Lorenz dominance relationship over the Pareto dominance relationship. For the comparison between L-MOACS and MOACS and that between L-MOACS and MOACS-GLS, the Lorenz dominance relationship presents more advantages compared with the Pareto dominance.

## 5. Conclusion

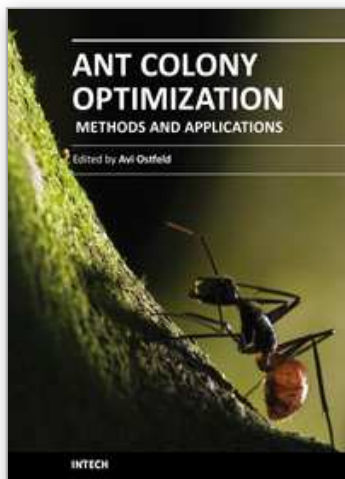
In this paper, we have studied a multiobjective buffers sizing problem using ant colony optimization. The two objectives of our study are the maximization of the throughput rate and the minimization of the total size of the buffers. To solve the problem, three multiobjective metaheuristics have been developed: a multiobjective ant colony optimization algorithm, a multiobjective ant colony algorithm with a guided local search and a Lorenz multiobjective ant colony optimization algorithm. Computational experiments have been realized on different assembly lines configurations and we have compared our three methods to each others. We have noticed that the L-MOACS performances are better than those of the MOACS and slightly better than those of the MOACS-GLS algorithm. Regarding the perspectives of this work, other methods based on the Pareto or the Lorenz dominance may be tested such as genetic algorithms or particle swarm optimization algorithms. Other hybridization techniques may also be tested in order to see the impact on the achieved solutions. Finally, exact methods may be developed to compare the optimal solutions with those of the applied metaheuristics.

## 6. References

- Abdul-Kader, W. (2006). Capacity improvement of an unreliable production line – an analytical approach. *Computers & Operations Research*, 33, 1695 – 1712.
- Altıparmak, A.; Bugak, A. & Dengiz, B. (2002). Optimization of buffer sizes in assembly systems using intelligent techniques. In *Proceedings of the 2002 Winter Simulation Conference*, pp. 1157–1162, December 2002, San Diego, USA.
- Benlian, X. & Zhiqian, W. (2007). A multi-objective-ACO-based data association method for bearings-only multi-target tracking. *Communications in Nonlinear Science and Numerical Simulation*, 12, 1360 - 1369.
- Chehade, H.; Amodeo, L.; Yalaoui, F. & De Guglielmo, P. (2007). Optimisation multiobjectif appliquée au problème de dimensionnement de buffers. In *Proceedings of the International Workshop on Logistic and Transport*, November 2007, Sousse, Tunisia.
- Chehade, H. ; Yalaoui, F.; Amodeo, L. & De Guglielmo, P. (2009). Optimisation multi-objectif pour le problème de dimensionnement de buffers. *Journal of Decision Systems*, 18, 2, 257–287.
- Collette, Y. & Siarry, P. (2002). *Optimisation multiobjectif*, Eyrolles, ISBN 2-212-11168-1, Paris.
- D'Souza, K. & Khator, S. (1997). System reconfiguration to avoid deadlocks in automated manufacturing systems. *Computers & Industrial Engineering*, 32, 2, 445 – 465.



- Dolgui, A.; Ereemeev, A.; Kolokolov, A. & Sigaev, V. (2002). A Genetic Algorithm for Allocation of Buffer Storage Capacities in a Production Line with Unreliable Machines. *Journal of Mathematical Modelling and Algorithms*, 1, 89 – 104.
- Dréo, J.; Pétrowski, A.; Siarry, P. & Taillard, E. (2003). *Métaheuristiques pour l'optimisation difficile*, Eyrolles, ISBN 2-212-11368-4, Paris.
- Dugardin, F.; Yalaoui, F. & Amodeo, L. (2009a). New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *European Journal of Operational Research*, 203, 1, 22 – 31.
- Dugardin, F.; Amodeo, L. & Yalaoui, F. (2009b). Multiobjective scheduling of a reentrant hybrid flowshop. In *Proceedings of CIE'39*, pp. 193 – 198, July 2009, Troyes, France.
- Hamada, M.; Martz, H.; Berg, E. & Koehler, A. (2006). Optimizing the product-based availability of a buffered industrial process. *Reliability Engineering and System Safety*, 91, 1039 – 1048.
- Han, M.S. & Park, D.J. (2002). Optimal buffer allocation of serial production lines with quality inspection machines. *Computers & Industrial Engineering*, 42, 75 – 89.
- Hani, Y.; Amodeo, L.; Yalaoui, F. & Chen, H. (2007). Ant colony optimization for solving an industrial layout problem. *European Journal of Operational Research*, 183, 633 – 642.
- Harris, J. & Powell, S. (1999). An algorithm for optimal buffer placement in reliable serial lines. *IIE Transactions*, 31, 287 – 302.
- Kilby, P.; Prosser, P. & Shaw, P. (1999). Guided local search for the vehicle routing problem with time windows. In: Voss S, Martello S, Osman IH, Roucairol C, editors. *Meta-heuristics advances and trends in local search paradigms for optimization*. Boston: Kluwer; 473 – 86.
- Kostreva, M. & Ogryczack, W. (1999). Linear optimization with multiple equitable criteria. *RAIRO Operations Research*, 33, 275 – 297.
- Kostreva, M.; Ogryczack, W. & Wierzbicki, A. (2004). Equitable aggregation and multiple criteria analysis. *European Journal of Operational Research*, 158, 2, 362 – 377.
- Lutz, C.; Roscoe Davis, K. & Sun, M. (1998). Determining buffer location and size in production lines using tabu search. *European Journal of Operational Research*, 106, 301 – 316.
- Nahas, N.; Ait-Kadi, D. & Nourelfath, M. (2006). A new approach for buffer allocation in unreliable production lines. *International Journal of Production Economics*, 103, 873 – 881.
- Papadopoulos, H.T. & Vidalis, M. (2001). Minimizing WIP inventory in reliable production lines. *International Journal Production Economics*, 70, 185 – 197.
- Pellegrini, P.; Favaretto, D. & Moretti, E. (2007). Multiple Ant Colony Optimization for a Rich Vehicle Routing Problem: A Case Study. *Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes In Computer Science*, 4693, 627 – 634.
- Perny, P.; Spanjaard, O. & Storme, L.X. (2006). A decision-theoretic approach to robust optimization in multivalued graphs. *Annals of Operations Research*, 147, 317 – 341.
- Riise, A. (2002). Comparing genetic algorithms and tabu search for multiobjective optimization. In *Proceedings of the IFORS conference*, July 2002, Edinburgh, UK.
- Spinellis, D.; Papadopoulos, C. & MacGregor Smith, J. (2000). Large production line optimisation using simulated annealing. *International Journal of Production Research*, 38, 3, 509 – 541.
- Voudouris, C. & Tsang, E. (1996). Partial constraint satisfaction problems and guided local search. In *proceedings of the second international conference on practical application of constraint technology (PACT'96)*, pp. 337 – 356, London.
- Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on evolutionary computation*, 3, 4, 257 – 271.



## **Ant Colony Optimization - Methods and Applications**

Edited by Avi Ostfeld

ISBN 978-953-307-157-2

Hard cover, 342 pages

**Publisher** InTech

**Published online** 04, February, 2011

**Published in print edition** February, 2011

Ants communicate information by leaving pheromone tracks. A moving ant leaves, in varying quantities, some pheromone on the ground to mark its way. While an isolated ant moves essentially at random, an ant encountering a previously laid trail is able to detect it and decide with high probability to follow it, thus reinforcing the track with its own pheromone. The collective behavior that emerges is thus a positive feedback: where the more the ants following a track, the more attractive that track becomes for being followed; thus the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. This elementary ant's behavior inspired the development of ant colony optimization by Marco Dorigo in 1992, constructing a meta-heuristic stochastic combinatorial computational methodology belonging to a family of related meta-heuristic methods such as simulated annealing, Tabu search and genetic algorithms. This book covers in twenty chapters state of the art methods and applications of utilizing ant colony optimization algorithms. New methods and theory such as multi colony ant algorithm based upon a new pheromone arithmetic crossover and a repulsive operator, new findings on ant colony convergence, and a diversity of engineering and science applications from transportation, water resources, electrical and computer science disciplines are presented.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hicham Chehade, Lionel Amodeo and Farouk Yalaoui (2011). Ant Colony Optimization For Multiobjective Buffers Sizing Problems, Ant Colony Optimization - Methods and Applications, Avi Ostfeld (Ed.), ISBN: 978-953-307-157-2, InTech, Available from: <http://www.intechopen.com/books/ant-colony-optimization-methods-and-applications/ant-colony-optimization-for-multiobjective-buffers-sizing-problems>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen