

Analysis and Adjustment of a Genetic Algorithm for Non-Permutation Flowshops*

Gerrit Färber
gerrit_faerber@gmx.de

Anna M. Coves Moreno
anna.maria.coves@upc.edu

Instituto de Organización y Control de Sistemas Industriales
Universidad Politécnica de Cataluña, Barcelona, Spain

First author (**for correspondence**):

Gerrit Färber

Email: *gerrit_faerber@gmx.de*

Tel: +34 93 401 17 06

Instituto de Organización y Control de Sistemas Industriales

Universidad Politécnica de Cataluña

Av. Diagonal 647, Planta 11, 08028 Barcelona, Spain

Second author:

Anna M. Coves Moreno

Email: *anna.maria.coves@upc.edu*

Tel: +34 93 401 66 53

Instituto de Organización y Control de Sistemas Industriales

Universidad Politécnica de Cataluña

Av. Diagonal 647, Planta 11, 08028 Barcelona, Spain

*This work is partially supported by the Ministry of Science and Technology of Spain, and the Funding for Regional Research DPI2004-03472

Abstract

The viability of many heuristic procedures strongly depends on the adequate adjustment of parameters. This work presents an adjustment procedure which was applied to a Genetic Algorithm. First, a preliminary analysis is performed, intended to obtain a better understanding of the behavior of the parameters, as for example to estimate how likely it is for the preceding adjustment of the parameters to remain in local minima. Special attention is paid on the variability of the solutions with respect to their repeatability. The four phases of the adjustment procedure are *Rough-Adjustment*, *Repeatability*, *Clustering* and *Fine Adjustment*.

Keywords: Parameter adjustment, Genetic Algorithm, Non-Permutation Flowshop, Assembly line

For a better understanding of the analysis and the adjustment of the Genetic Algorithm (GA) first the specific problem in the area of sequencing in a non-permutation flowshop is introduced to which the GA is applied, as well as the algorithm itself.

In the classical production line, only products with the same options were processed at once. Products of different models, providing distinct options, were either processed on a different line or major equipment modifications were necessary. For today's production lines this is no longer desirable and more and more rise the necessity of manufacturing a variety of different models on the same line, motivated by offering a larger variety of products to the client. Furthermore, the stock for finished products is reduced considerably with respect to a production with batches, and so are the expenses derived from it. Mixed model production lines consider more than one model being processed on the same production line in an arbitrary sequence. Nevertheless, the majority of publications in this area are limited to solutions which determine the job sequence before the jobs enter the line and maintain it without interchanging jobs until the end of the production line, known as permutation flowshop.

In Potts et al., 1991, and Liao et al., 2006, studies of the benefit of using non-permutation flowshops are presented. Furthermore, there exist various designs of production lines which permit resequencing of jobs: using large buffers (Automatic-Storage-and-Retrieval-System) which decouple one part of the line from the rest of the line Lee and Schaefer, 1997; buffers which are located off-line Lahmar et al., 2003; hybrid or flexible lines Engström et al., 1996; and more seldomly, the interchange of job attributes instead of physically changing the position of a job within the sequence Rachakonda and Nagane, 2000. Resequencing of jobs on the line is even more relevant with the existence of an additional cost or time, occurring when at a station the succeeding job is of another model, known as setup-cost and setup-time Bolat, 1994.

The present work considers a flowshop with the possibility to resequence jobs between consecutive stations. The problem is NP-hard, see Garey et al., 1976, and as highlighted by Lahmar et al., 2003, only few resequencing possibilities are necessary in order to achieve the greatest benefit. Furthermore, the buffers are located off-line either accessible from a single station (intermittent case) or from various stations (centralized case). In both cases, it is considered that a job may not be able to be stored in a buffer place, due to its extended physical size, see figure 1.

The considered problem is relevant to various flowshop applications such as chemical productions dealing with client orders of different volumes and different sized resequencing tanks. Also in productions where split-lots are used for engineering purpose, such as the semiconductor industry. Even in the production of prefabricated houses with, e.g., large and small walls passing through consecutive stations where electrical circuits, sewerage, doors, windows and isolation are applied.

After the design of the GA which is used to solve the concerned problem, a preliminary analysis is performed, intended to obtain a better understanding of the behavior of the parameters used by the applied Genetic Algorithm (GA), as for example to estimate how likely it is for the succeeding adjustment of the parameters to remain in local minima. Special attention is paid on the variability of the solutions with respect to their repeatability. Then, based on the preliminary analysis, an adjustment procedure is

proposed which adequately adjusts the parameters, including differently sized problems. The four phases of the adjustment procedure are *Rough-Adjustment*, *Repeatability*, *Clustering* and *Fine Adjustment*. In what follows the problem is formulated with more detail and the applied Genetic Algorithm is described. Thereafter, the accomplished performance study is presented and finally conclusions are presented which are already useful at the time a production line is being designed.

1 Problem Definition

The realized work is based on the classical flowshop in which the jobs ($J_1, J_2, \dots, J_j, \dots, J_n$) pass consecutively through the stations ($I_1, I_2, \dots, I_i, \dots, I_m$). Furthermore, after determined stations, off-line buffers B_i permit to resequence jobs. The buffer provides various buffer places ($B_{i,1}, B_{i,2}, \dots$) and each buffer place is restricted by the physical size of the jobs to be stored. As can be seen in figure 1a, job J_2 can be stored in buffer place $B_{i,1}$ as well as in $B_{i,2}$. Whereas, the next job J_3 can be stored only in buffer place $B_{i,2}$, because of the physical size of the job exceeding the physical size of the buffer place $B_{i,1}$, see figure 1b.

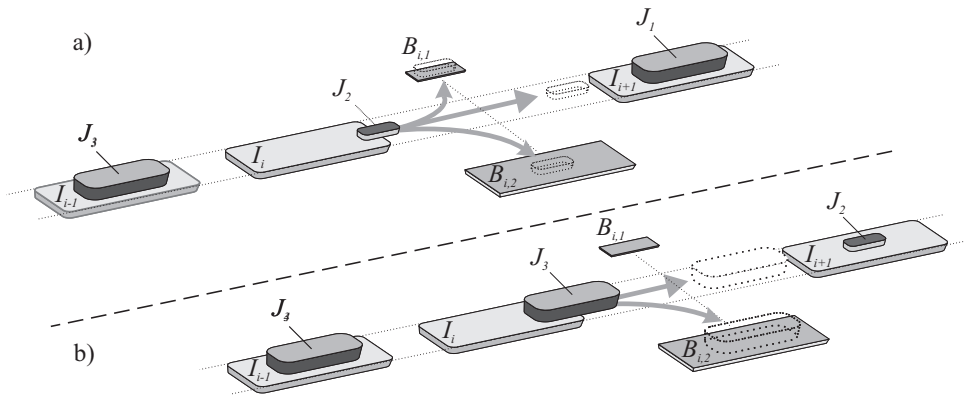


Figure 1: **Scheme of the considered flowshop.** The jobs J_j pass consecutively through the stations I_i . The buffer B_i permits to temporally store a job with the objective of reinserting it at a later position in the sequence. a) Job J_2 can pass through any of the two buffer places $B_{i,1}$ or $B_{i,2}$ of buffer B_i . b) Job J_3 can pass only through buffer place $B_{i,2}$, due to its physical size.

In a first step, the resequencing buffers are located intermittent, between two consecutive stations. In this case the buffer is assigned to the precedent station and may be accessed only by this station. Then, for an additional benefit, a single resequencing buffer is used, with access from various stations, while the limitations on the physical size of the buffer places are maintained.

2 Genetic Algorithm (GA)

The concept of the GA was first formulated by Holland, 1973, and Holland, 1975, and can be understood as the application of the principles of evolutionary biology, also known as the survival of the fittest, to computer science. Genetic algorithms are typically implemented as a computer simulation in which a population of chromosomes, each of which represents a solution of the optimization problem, evolves

toward better solutions. The evolution starts from an initial population which may be determined randomly. In each generation, the fitness of the whole population is evaluated and multiple individuals are stochastically selected from the current population, based on their fitness and modified to form a new population. The alterations are biologically-derived techniques, commonly achieved by inheritance, mutation and crossover. Multiple Genetic Algorithms were designed for mixed model assembly lines such as Bolat et al., 2005, Levitin et al., 2006, Wang et al., 2006, and Rubén and Concepción, 2006.

The heuristic used here is a variation of the GA explained in Michalewicz, 1996. The genes represent the jobs which are to be sequenced. The chromosomes v , determined by a series of genes, represent a sequence of jobs. A generation is formed by R chromosomes and the total number of generations is G . In the permutation case, the size of a chromosome is determined by the number of jobs, the fraction Π . In the non-permutation case, the chromosomes are $L + 1$ times larger, resulting in the fractions $\Pi'_1, \dots, \Pi'_{L+1}$, being L the number of resequencing possibilities. In both cases, special attention is required when forming the chromosomes, because of the fact that for each part of the production line every job has to be sequenced exactly one time.

The relevant information for each chromosome is its fitness value (objective function), the number of job changes and the indicator specifying if the chromosome represents a feasible solution. A chromosome is marked unfeasible and is imposed with a penalty, if a job has to be taken off the line and no free buffer place is available or the physical size of the job exceeds the size limitation of the available buffer places. When two solutions result in the same fitness, the one with fewer job changes is preferred.

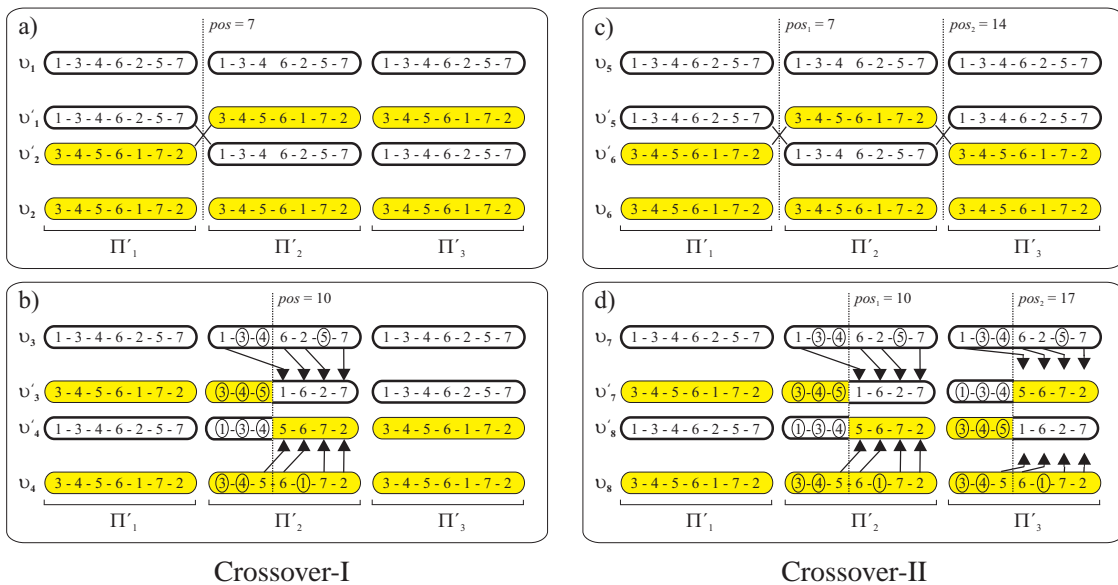


Figure 2: **Operators crossover-I and crossover-II.** a) and c) In the simple case the crossing takes place between two main fractions of the chromosome. After the crossover point the chromosomes are completely crossed over. b) and d) In the more complex case it has to be assured that each job is sequenced exactly one time for each fraction of the chromosome.

2.1 Genetic operators

The genetic operators specify in which way the subsequent population is generated by reproduction of the present population, taking into account that "fitter" solutions are more promising and therefore are more likely to reproduce. Even an unfeasible solution is able to reproduce, because of the fact that it may generate valuable and feasible solutions in one of the following generations. The used genetic operators are inheritance, crossover and mutation. The value p_X is the percentage with which a genetic operator X is applied to a chromosome.

Inheritance: This operator is determined by two parameters. The parameter MBS determines the percentage of the best solutions which will be copied directly to the next generation, called the cluster of promising chromosomes, and ensures that promising chromosomes are not extinct. Then, in order to not remain in a local minimum, the parameter p_b determines the percentage of chromosomes which are removed from this cluster.

Crossover: This operator specifies the operation of interchanging information of two chromosomes. Two crossover operations are applied, crossover-I (figure 2a,b) and crossover-II (figure 2c,d). The probabilities with which these operations are applied to a chromosome are p_{c-I} and p_{c-II} , and the crossover points are defined by the random number pos , and the pair pos_1 and pos_2 , respectively.

If the crossover point (pos , pos_1 and pos_2) is a multiple of the number of jobs to be sequenced, the crossover operation is simple and takes place between two main fractions of the chromosome, i.e. after the crossover point the chromosomes are completely crossed over. Whereas, in the complex case the crossover points are located within a main fraction of the chromosome and it has to be assured explicitly that each job is sequenced exactly one time for each fraction of the chromosome.

Mutation: This operator specifies the operation of relocating jobs at position pos_1 to position pos_2 within the same fraction of a chromosome. Two mutation operators are applied, mutation-I and mutation-II (figure 3). Furthermore, there exist two cases for mutation-I: forward mutation, where $pos_1 < pos_2$; and backward mutation, where $pos_1 > pos_2$. In the first case, a single job has to be taken off the line, and in the second case, in order to let a single job pass, a group of succeeding jobs has to be taken off the line, resulting in a larger effort to realize. The probabilities of this operator are $p_{m-I(f)}$, $p_{m-I(b)}$ and p_{m-II} .

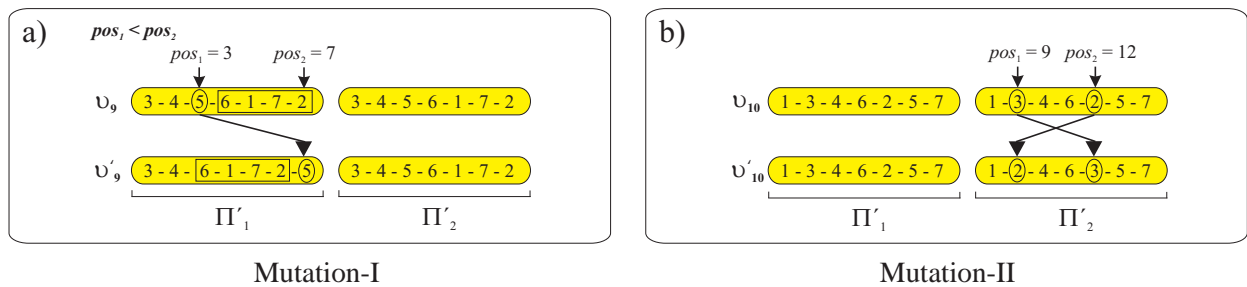


Figure 3: **Operators mutation-I and mutation-II.** a) The job at position pos_1 is taken off the line and reinserted to the line at position pos_2 . b) The two jobs at position pos_1 and pos_2 are interchanged.

2.2 Cascade of the GA

In the case in which the considered flowshop permits resequencing between stations, the size of the chromosomes increases by a factor, linear to the number of stations with access to resequencing buffers. In order to further enhance the GA, the proceeding is divided in two steps. In a first step the GA ignores the possibility of resequencing jobs within the production line and therefore considers only permutation sequences as possible solutions. The chromosome is reduced to the number of jobs N . The last generation leads to a preliminary best generation, the worst solution of this generation is replaced by the so far best solution.

Permutation sequences (step-1):

- Resequencing does not take place.
- Chromosome-size is only depending on the number of jobs, N .
- Best generation forms the initial generation of step 2.

In a second step the GA takes into account the resequencing possibilities provided by stations with access to resequencing buffers. The chromosome-size is depending on the number of jobs to be processed, N , and the number of stations with access to resequencing buffers, L . The initial solutions which form the initial generation consist of permutation sequences. The permutation sequences Π'_1 (for v_1 to v_R) are copied to each of the sequences Π'_1 to Π'_{L+1} (for v_1 to v_R).

Non-permutation sequences (step-2):

- Resequencing takes place between determined stations.
- Chromosome-size is depending on the number of jobs to be processed, N , and the number of stations with access to resequencing buffers, L .

3 Preliminary Analysis and Adjustment of Parameters

Prior to the adjustment of the parameters of the GA, a preliminary analysis is performed, intended to obtain a better understanding of the behavior of the parameters, as for example to estimate how likely it is for the succeeding adjustment of the parameters to remain in local minima.

3.1 Preliminary analysis

In the preliminary analysis the two parameters for crossover and the three parameters for mutation are studied by varying them pairwise in the range of 0.0 to 0.1. The parameters which are not considered variables, are listed in table 1, together with their set values. Each experiment is repeated with a set of four seeds, then the average of the objective function is used to plot the results.

Name	Description	Values
N	Number of jobs	10
M	Number of stations	5
	Stations with access to resequencing buffers	2, 4
D	Number of buffer places for each buffer	3
R	Population size	100
G	Number of Generations	1000
MBS	Number of best solutions to maintain	5
p_b	Probability to eliminate best solutions	0.1
FP	Penalty for non-feasible solution	10
Seed	Seed for generation of random number	47, 57, 67, 77

Table 1: Fixed parameters used for the preliminary analysis of the GA.

Crossover-I versus Crossover-II

The first analysis varies the two parameters crossover-I and crossover-II. The sum of the two parameters may not be larger than 1.0, therefore only values are considered which do not exceed this limit. The three remaining parameters for mutation are set to low values ($p_{m-I(f)} = 0.1$, $p_{m-I(h)} = 0.3$, $p_{m-II} = 0.1$).

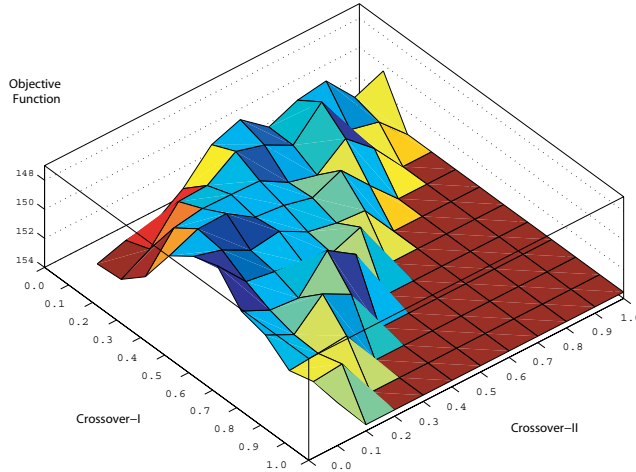


Figure 4: Study of the influence of p_{c-I} , the probability for crossover-I and p_{c-II} , the probability for crossover-II, on the objective function.

Figure 4 shows the influence of the two parameters for crossover on the resulting value of the objective function. In order to plot the mesh of solution points, the combinations of the two parameters which result in the sum being larger than 1.0, are complemented with the maximum value of all considered points. This results in a plane surface for the half of the plotted mesh without any relevance for the analysis. The plotted mesh shows that a fairly continuous surface is formed with a wide valley. The interpretation of this valley leads to the conclusion that neither setting the considered parameters at the same time to small values, nor setting them such that their sum is close to 1.0 is advisable.

In the succeeding analysis the parameters for crossover-I and crossover-II are set to 0.3 and 0.2, respectively. The selection of these values is a combination which results to be within the valley.

Mutation-I (f) versus Mutation-I (b)

The second analysis varies the two parameters mutation-I (f) and mutation-I (b). The two parameters do not face any limitation amongst them. The parameter for mutation-II, which is not analyzed yet, is set to a low value ($p_{m-II} = 0.1$). Figure 5 shows the influence of the two parameters for mutation-I on the resulting value of the objective function. The plotted mesh shows that a continuous surface is formed which is improving when both parameters for mutation-I are decreased. Except for the case in which both of them become too small. In the following analysis of mutation-I (b) is set to 0.1.

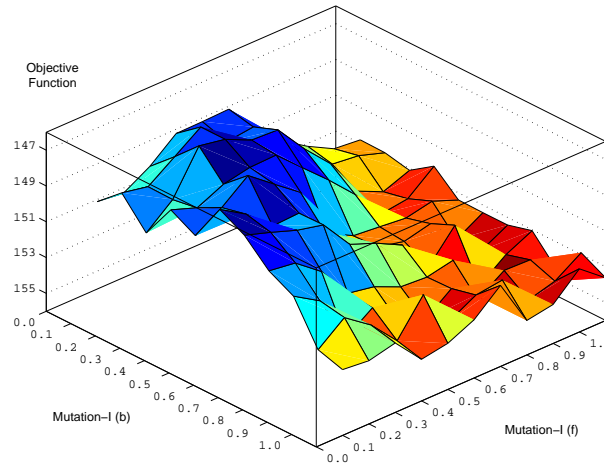


Figure 5: Study of the influence of $p_{m-I(f)}$, the probability for mutation-I (f) and $p_{m-I(b)}$, the probability for mutation-I (b), on the objective function.

Mutation-I (f) versus Mutation-II

The third analysis varies the two parameters mutation-I (f) and mutation-II. The two parameters do not face any limitation. Figure 6 shows the influence of the two parameters mutation-I (f) and mutation-II on the resulting value of the objective function. Similar to the previous case, the plotted mesh shows that a continuous surface is formed which is improving, when both parameters are decreased.

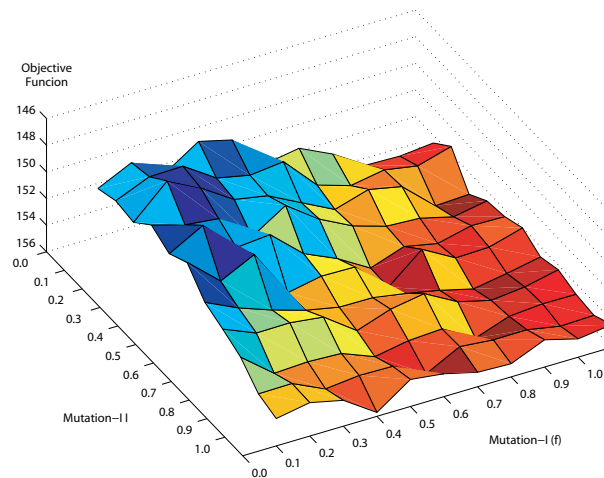


Figure 6: Study of the influence of p_{m-II} , the probability for mutation-II and $p_{m-I(f)}$, the probability for mutation-I (f), on the objective function.

Variability of solutions

The GA is based on random numbers, giving the algorithm its strength. However, this also leads to the disadvantage that the algorithm on the other hand is not very predictable and in order to determine promising parameters, useful for a multitude of problems, the analysis of the parameters is to be repeated with various different seeds.

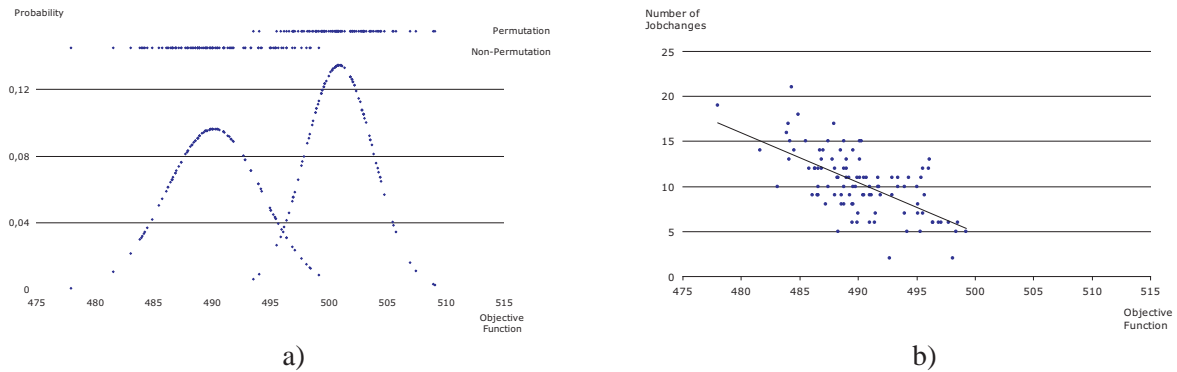


Figure 7: Variability of the solutions of the Genetic Algorithm. a) The same problem is solved 100 times, each time a different seed is used, for the permutation case as well as for the non-permutation case. b) Dependency of the number of job changes on the objective function

The variability of the GA with respect to the obtained solutions is shown in figure 7. The same problem is solved by the GA 100 times, each time a different seed is used, for the permutation case as well as for the non-permutation case. The solutions, permitting non-permutation sequences, in general result in better solutions with a larger deviation. The average value of the objective function of the particular example, used in figure 7a, is 500.93 with a standard deviation of 2.96 for the permutation case and 490.21 with a standard deviation of 4.13 for the non-permutation case. Analyzing the obtained data with respect to the number of job changes with more detail, figure 7b shows that in order to obtain better solutions, the number of jobs, which have to be taken off the line for resequencing, tends to be higher.

Synopsis of preliminary analysis

Concluding the preliminary analysis of the parameters, it can be resumed that the variation of the five considered parameters for crossover and mutation result in a continuous solution space with smooth transitions. It can be outlined that it is expected that the adjustment of these five parameters will lead to values that can be found in the lower or in the mid part of their range. Furthermore, due to the variability of the results, it is necessary to repeat the individual experiments with several seeds in order to ensure a robust parameter adjustment.

3.2 Parameter adjustment

Previous to the use of the Genetic Algorithm, the parameters which result in a multitude of possible combinations, have to be adjusted. In what follows and based on the preliminary analysis, an adjustment proceeding is proposed in order to adjust the parameters which lead to rapid convergence and encouraging solutions.

Rough adjustment

On account of the numerous parameters of the presented GA, it is necessary to roughly adjust the parameters by using a predetermined number of discrete values for each parameter, see table 2. The total number of possible sets of parameters is 3456 and only one seed is used.

Name	Description	Range	Values
R	Number of parents	Small, Medium	70, 100
MBS	Number of best solutions to maintain	Medium	5
p_b	Probability eliminate best solutions	Small, Medium	0.1, 0.4
p_{C-I}	Probability of Crossover-I	Small, Medium, Large	0.1, 0.3, 0.6 ¹
p_{C-II}	Probability of Crossover-II	Small, Medium, Large	0.1, 0.3, 0.6 ¹
$p_{m-I(f)}$	Probability of Mutation-I (forward)	Small, Medium, Large	0.1, 0.3, 0.5
$p_{m-I(b)}$	Probability of Mutation-I (backward)	Small, Medium, Large	0.1, 0.3, 0.5
p_{m-II}	Probability of Mutation-II	Small, Medium, Large	0.1, 0.3, 0.5
FP	Penalty for non-feasible solution	Small, Medium	10, 50

Table 2: Discrete values of the parameters for the rough adjustment of the GA.

In order to adjust the parameters of the GA in a robust manner, the above mentioned set of parameters is applied to a series of different sized problems, first for instances which consider permutation sequences and then for instances which consider non-permutation sequences. The size of the problems is determined by N , the number of jobs to be sequenced, M , the number of stations and in the non-permutation case by L , the number of resequencing possibilities. Table 3 and 4 show the considered variations in problem size which lead to 14 differently sized problems.

Name	Description	Range	Values
N	Number of jobs	Small, Medium	10, 20
$M-L$	Number of stations - number of stations with access to resequencing buffers	Small - Null, Medium - Null Large - Null	5-0 10-0 20-0

Table 3: Variable parameters of the flowshop used for the analysis and adjustment of the GA (Permutation case).

Name	Description	Range	Values
N	Number of jobs	Small, Medium	10, 20
$M-L$	Number of stations - number of stations with access to resequencing buffers	Small - Null, Small - Small Medium - Null Medium - Small	5-2 10-2 20-2 20-5

Table 4: Variable parameters of the flowshop used for the analysis and adjustment of the GA (Non-permutation case).

The sets of parameters are then summarized and the 300 most promising ones, which show good performance on all 14 different problem sizes are used for further analysis.

¹ $p_{C-I} + p_{C-II}$ can not exceed 1.0, therefore the combination $p_{C-I} = 0.6$ and $p_{C-II} = 0.6$ is not considered.

Repeatability

The use of only one seed in the rough adjustment requires to determine amongst the promising parameter sets which one achieves good results for a multitude of seeds. The fact that a parameter set achieves good results for different seeds indicates that the same parameter set also performs well for different flowshops. The promising sets of parameters are taken and verified with 16 different seeds on the 14 differently sized problems.

Clustering of parameter sets

Once the sets of promising parameters are examined with respect to repeatability, one set is used for the fine adjustment. The Matlab toolbox from Balasko et al., 2005, is used to group the parameter sets into clusters and use the one which performs best.

Fine adjustment

Due to the fact that in the previous analysis predetermined discrete values for the parameters are used, a fine adjustment succeeds. The parameters for crossover and mutation are subject to an adjustment of two times 0.05 for the previously determined sets of parameters and are revised with 16 seeds on the 14 differently sized problems, used for the repeatability.

Name	Description	Non-Perm	Perm
R	Number of parents	100	100
MBS	Number of best solutions to maintain	5	5
p_b	Probability eliminate best solutions	0.1	0.4
p_{c-I}	Probability of Crossover-I	0.3	0.5
p_{c-II}	Probability of Crossover-II	0.6	0.35
$p_{m-I(f)}$	Probability of Mutation-I (forward)	0.25	0.1
$p_{m-I(b)}$	Probability of Mutation-I (backward)	0.25	0.1
p_{m-II}	Probability of Mutation-II	0.25	0.1
FP	Penalty for non-feasible solution	10	10

Table 5: Adjustment of variable parameters of the GA obtained by the extended analysis.

Due to the cascade of the GA, section 2.2, the proposed proceeding is divided into two cases, the permutation case and the non-permutation case. As a result, two different sets of parameters are obtained. The adjusted parameters are listed in table 5.

4 Performance Study

For the study of performance, a flowshop which consists of 5 stations is considered. The range of the production time is $[0...20]$ such that for some jobs exists zero-processing time at some stations, for the setup-cost $[2...8]$ and for the setup-time $[1...5]$. The number of jobs is varied from 5 to 100 with increments of 5. The objective function, is the weighted sum of the makespan (factor of 1.0) and the setup-cost (factor of 0.3), where the setup-time is not concerned with a weight but is indirectly included in the calculation of the makespan.

Case	Intermittent			Centralized		
	l	m	s	l	m	s
(300)	1/2	0/0	0/0	3	0	0
(111)	0/1	1/0	0/1	1	1	1
(102)	0/1	0/0	1/1	1	0	2
(012)	0/0	0/1	1/1	0	1	2

Table 6: Allocation of the buffer places to the buffers. In the intermittent case the allocation is done to two different buffers.

Introducing limitations on the physical size of the buffer places on one side restricts possible solutions but on the other side minimizes the necessary buffer area. This limitation arises, for example, in a chemical production. The arrangement of two tanks which are located off the line, accessible after a certain station, equals an intermittent resequencing buffer with two buffer places. With tank capacities of 50 and 100 liters, a client order of 80 liters can be stored only in the larger of the two tanks which is capable of storing this volume. Whereas, a client order of 50 liters can be stored in either of the tanks. A close look at the local conditions may amortize an increase in the objective function compared to an investment reduction with respect to tank size and gained area.

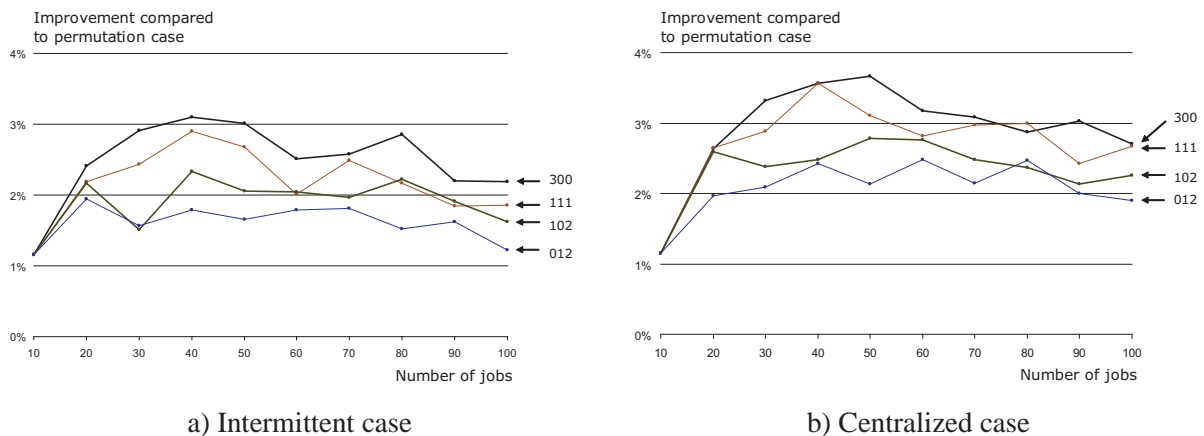


Figure 8: Influence of the variation of the physical size of the buffer places. "102" represents 1 large, 0 medium and 2 small buffer places. In the intermittent case, the buffer places are divided to two buffers, each with access from a designated station. In the centralized case, the same two stations have simultaneously access to the buffer, containing all three buffer places. The ratio of jobs is $\frac{3}{10}$ large, $\frac{3}{10}$ medium and $\frac{4}{10}$ small.

As a concrete example, three differently sized buffer places (large, medium, small) are available and the ratio of jobs is $\frac{3}{10}$ large, $\frac{3}{10}$ medium and $\frac{4}{10}$ small. As in the previous section, the second and the third station have access to the resequencing buffers and table 6 shows the allocation of the buffer places to the buffers, considering eight scenarios. The code "300" represents 3 large, 0 medium and 0 small buffer places. In the intermittent case the first buffer is provided with 1 and the second buffer with 2 large buffer places. In the centralized case the same two stations have access to a single centralized buffer, containing the three buffer places. Figure 8 shows the influence of the limitation of the physical size. The variation of the size of the buffer places towards smaller buffer places on the one hand decreases the benefit achieved by the possibility of resequencing jobs. On the other hand, it may amortize when taking into account the reduction of investment with respect to tank size and gained area.

5 Conclusions

This paper has presented a preliminary analysis and four step adjustment procedure for a Genetic Algorithm which was applied to a mixed model non-permutation flowshop using constrained buffers. The algorithm uses the genetic operators inheritance, crossover and mutation and is designed to consider intermittent or centralized resequencing buffers. Furthermore, the buffer access is restricted by the number of buffer places and the physical size of jobs. The presented GA uses a two step cascade, first seeking permutation sequences, then widening the solution space to non-permutation sequences.

The preliminary analysis of the behavior of the genetic operators appeared to be valuable, showing that the solution space, when varying two genetic operators, is a continuous surface without local minima. Furthermore the variability of the solutions was observed, pointing out that the solutions permitting non-permutation sequences lead to better solutions but with a larger deviation. The four step parameter adjustment was designed such that both, flexibility and robustness were ensured. The four phases of the adjustment procedure are: *Rough-Adjustment*, considering a widespread parameter set with the utilization of a single seed for the generation of the random numbers; *Repeatability*, followed by *Clustering*, ensuring the good performance of the promising parameter sets; and finally *Fine Adjustment*, further improving the selected parameter set.

The adjustment procedure resulted in two parameter sets: one for the first cascade of the GA, which only permits permutation sequences, and one for second cascade, permitting resequencing of jobs within the production line, non-permutation sequences.

Then, the study of performance demonstrated the effectiveness of resequencing jobs within the line. The results of the simulation experiments revealed the benefits that come with a centralized buffer location, compared to the intermittent buffer location. It either improves the solution or leads to the utilization of fewer resequencing buffer places. An increased number of large buffer places clearly improves the objective function and including buffers, constrained by the physical size of jobs to be stored, on one side limits the solutions but on the other side minimizes the necessary buffer area.

In order to take full advantage of the possibilities of resequencing jobs in a mixed model flowshop, additional installations may be necessary to mount, like buffers, but also extra efforts in terms of logistics complexity may arise. The additional effort is reasonable if it pays off the necessary investment. Due to the strong dependency on local conditions, a general validation is not simple and was not part of this work.

References

- Balasko, B., Abonyi, J., and Feil, B.(2005).** Fuzzy clustering and data analysis toolbox; for use with matlab.
- Bolat, A.(1994).** Sequencing jobs on an automobile assembly line: objectives and procedures. *International Journal of Production Research*, 32(5):1219–1236.
- Bolat, A., Al-Harkan, I., and Al-Harbi, B.(2005).** Flow-shop scheduling for three serial stations with the last two duplicate. *Computers & Operations Research*, 32(3):647–667.
- Engström, T., Jonsson, D., and Johansson, B.(1996).** Alternatives to line assembly: Some swedish examples. *International Journal of Industrial Ergonomics*, 17(3):235–245.
- Garey, M., Johnson, D., and Sethi, R.(1976).** The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1:117–29.
- Holland, J.(1973).** Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.*, 2(2):88–105.
- Holland, J.(1975).** Adaptation in natural and artificial systems. *University of Michigan Press, Ann Arbor*, page s.
- Lahmar, M., Ergan, H., and Benjaafar, S.(2003).** Resequencing and feature assignment on an automated assembly line. *IEEE Transactions on Robotics and Automation*, 19(1):89–102.
- Lee, H. and Schaefer, S.(1997).** Sequencing methods for automated storage and retrieval systems with dedicated storage. *Computers and Industrial Engineering*, 32(2):351–362.
- Levitin, G., Rubinovitz, J., and Shnits, B.(2006).** A genetic algorithm for robotic assembly line balancing. *European Journal of Operational Research*, 168:811–825.
- Liao, C., Liao, L., and Tseng, C.(2006).** A performance evaluation of permutation vs. non-permutation schedules in a flowshop. *International Journal of Production Research*, 44(20):4297–4309.
- Michalewicz, Z.(1996).** *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 3rd edition.
- Potts, C., Shmoys, D., and Williamson, D.(1991).** Permutation vs. non-permutation flow shop schedules. *Operations Research Letters*, 10(5):281–284.
- Rachakonda, P. and Nagane, S.(2000).** Simulation study of paint batching problem in automobile industry. <http://sweb.uky.edu/~pkrach0/Projects/MFS605Project.pdf>. consulted 14.07.2004.
- Rubén, R. and Concepción, M.(2006).** A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169(3):781–800.
- Wang, L., Zhang, L., and Zheng, D.(2006).** An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*. Article in Press.