Two Heuristics for the Improvement of a Two-Phase Optimization Method for Manufacturing Systems*

Diego Rodríguez
Univ. of Zaragoza
c/María de Luna 3
Zaragoza 50018 Spain
dierodri@unizar.es

Armin Zimmermann
Techn. Univ. Berlin
Einsteinufer 17, Sekr. E-N 10
10587 Berlin Germany
azi@cs.tu-berlin.de

Manuel Silva
Univ. of Zaragoza
c/María de Luna 3
Zaragoza 50018 Spain
silva@unizar.es

Abstract – *The Optimization of Manufacturing Systems is computationally expensive in most cases. A meta-heuristic (Simulated Annealing) is considered here to control the overall optimization process. Stochastic Petri nets are used for the modelling and evaluation part. The basic idea is to split the optimization in two phases. In the first one a "near" optimal parameter set is quickly computed, which is improved in a second phase. This strategy has shown its ability to reduce the computational effort substantially in some cases in previous papers [10, 11, 12]. Several additional heuristics are developed in this work which aim at reducing the optimization effort even further. In a first improvement, the results of the approximation phase are analyzed further to gain deeper knowledge about the optimization parameter space. This knowledge is then used to control the algorithm parameters of the second optimization phase. The solutions obtained with these new techniques are comparable to the ones obtained in the original two phase optimization work, but the computational effort is reduced by 50 percent on average. In a second approach a new optimization scheme is proposed, which can be applied to models for which the fast approximation technique used in the two-phase approach cannot be used. This scheme takes advantage of the possibility of executing parameterized simulations of the Petri Net models.*

Keywords: *Manufacturing systems, Modeling, Petri Nets, Optimization.*

1 Introduction

The optimization of Manufacturing Systems is a complex problem to solve and only for a few simple examples it is possible to obtain good solutions in a reasonable amount of time. Namely, for realistic models of complex manufacturing systems, well-known direct optimization methods can not be used.

We use Petri nets (PNs) as the modeling paradigm for manufacturing systems. The use of Petri nets allows to model systems with intricate interleaving of cooperation and competition, thanks to the ability of nets to model conflicts and synchronizations. Provided with appropriately interpreted extensions, PNs lead to different formalisms useful in the different phases of the life-cycle of the system under design or operation (global modeling, performance evaluation, correctness analysis, implementation, scheduling, monitoring,...), constituting a formal modeling *paradigm* [2, 5].

The optimization problems solved here are related to the design variables of manufacturing systems. For example, the number of buffer places or the speed of a certain machine/AGV are variables that can change during the optimization process. These variables can be real, integers and even logical. The optimization function typically contains several terms. Some terms are related to the benefits that the production of a certain finished product produces. In addition there are other costs related for example, to the work in process or the constant costs or the cost corresponding to the investment in new machines, space in buffers, etc.

The utilization of meta-heuristic optimization techniques, like Genetic Algorithms [6], Simulated Annealing [1] or Tabu Search [8] for the derivation of solutions is a promising approach, but requires an underlying evaluation of performance measures for each considered parameter set. The main problem of these techniques appears when the system requires a computationally expensive evaluation in each optimization step. When the size of the parameter space is large in addition, the overall effort becomes unacceptable.

Throughout this paper the ASA package [7] that implements the Simulated Annealing method is used, but the heuristics do not depend on the optimization method and could easily be adopted for different ones. The paradigm used for the modelling and evaluation of the performance measures are stochastic Petri nets. Evaluation of the different performance measures is done with the Petri net tool TimeNET [9].

The application examples used here come from the area of manufacturing systems, but the presented techniques are well applicable to other domains in which

*This work has been partially granted by the CICYT project TAP. 98-0679

Petri nets are used. The complexity of the two models presented in this paper forbids the evaluation with direct numerical analysis due to the large size of the state space. Simulation with explicit confidence control is thus applied to compute the profit function values.

2 Previous Approaches

In a previous work [10], the use of a cache in the ASA algorithm decreased drastically (to some 15%) the number of solutions evaluated by the Petri Net software package. For an additional gain in the computational effort without losing quality of the solutions, the optimization process was then divided in two phases (pre and fine optimization) [11]. The pre-optimization phase typically takes only a small portion of the overall computation time spent. The approximate computation of the value of the cost (or profit) function for a certain parameter set was based on some performance bounds for the Petri Net model obtained from a set of linear programming problems [3, 4]. The use of the pre-optimization technique has proven efficient to decrease the computational effort of the complete optimization process to the fifth part in many cases [12]. However, there are more complex examples for which the optimization computational time to achieve a good quality solution was still too high. The heuristics presented in this paper try to cope with some of the problems that remained open in previous works. The two main issues tackled are:(1) to reduce the computational effort of the second phase of the optimization algorithm, and (2) to develop efficient optimization techniques for models in which the performance bounds technique is not applicable.

The two examples proposed here with their Petri Net models and their respective optimization problems can be checked in [12]. The first example is an assembly line with five machines. Three different parts A, B and C are assembled for one final product. Customer demands and waiting times are also considered, and one of the optimisation goals is to find the *best production policy* out of three classical manufacturing control strategies ("push", "on demand" and "kanban").

The second example is an FMS where two types of products A and B have to be produced. Parts of type A can first be processed by one of two machines. A manual operation and an assembly of an additional part have to follow, before the product is finished. B-type parts are first processed by machine 1. Afterwards they are tested at one manual work place. Parts that have been correctly processed are transported to the assembly station. After an assembly operation the product is finished. However, statistically it is known that 5% of the parts have to be reworked at machine 1, which is detected at the manual work place.

3 First Approach

The development of alternative strategies for the second phase of the optimization process is the key issue of this section. The need for new reduction of efforts in the optimization of manufacturing systems is discussed and two heuristic strategies for this problem are explained.

In previous papers [11, 12], we have shown that using a pre-optimization phase, the obtained results are good while the computation time could be reduced substantially. Most of the remaining time is spent in the fine optimization phase. In the original two phase scheme the only information transferred between the two phases is the parameter set for which the best solution was obtained in the pre-optimization phase. The heuristics that are presented in this paper follow the idea of analyzing the results of the first phase more thoroughly, in order to adjust the parameters of the second phase and thus make it faster. We try to reduce the number of simulation runs necessary during the second phase, because that is the most important influence. We propose two different strategies in order to decrease the effort spent in this fine optimization phase. Nevertheless as examples show, one must be careful not to restrict the second phase too much, because then the algorithm might not be able to come close to the real optimum (i.e. the quality of the solution will suffer).

The subsequent subsections explain the two strategies for reducing the computational effort. They use a reduction of the search space according to the quality of the first-phase solutions as well as smaller temperature parameters of the simulated annealing variables. In that way information from the first phase is used to decrease the remaining search space and to come to a solution faster. The results in terms of the computation effort and solution quality are given and compared for all presented techniques in section 5.

3.1 Search Space Reduction Strategy

The method consists in analyzing the "most promising" solutions during the first optimization phase. Parameter sets are considered promising if their corresponding profit value differs from the best one only by a certain relative percentage that will vary from 5 to 50 percent. The algorithm starts after the first phase has been finished and uses a constant *AccPerc*, which defines the acceptance interval. Every solution visited in the first phase is analyzed and if the value of the profit is in the acceptance interval we consider this solution as "accepted". For every parameter the maximum and minimum values contained in the accepted set are computed. The search space of the second optimization phase is then reduced to parameter sets that lie within this new maximum/minimum domain. This search space is an n-dimensional parallelepiped.

The *AccPerc* values considered here for the two examples are 5, 10, 20, 30 and 50 percent. The method is

	Parameter 1		Parameter 2		Parameter 3		Parameter 4	
	Min	*Max*	*Min*	*Max*	*Min*	*Max*	*Min*	*Max*
In. (100%)	1	10	1	10	1	9	1	3
50%	1	10	1	10	1	1	1	3
30%	1	9	1	10	1	1	1	3
20%	1	9	1	10	1	1	1	3
10%	1	9	1	10	1	1	1	2
5%	1	9	1	10	1	1	2	2

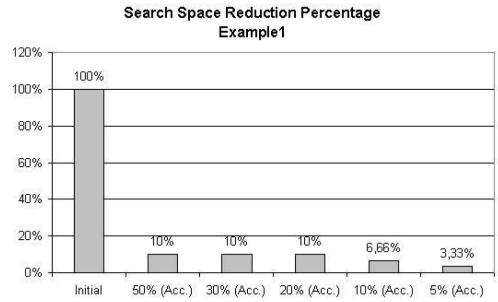

Figure 1: Search Space Reduction Strategy and Gain Percentage for the Assembly System – Example 1 in [12]

less restrictive for bigger values.

In Figure 1, the results obtained using the search space reduction method for the assembly system example [12] are shown. The upper table shows the maximum and minimum values of the parameters included in the optimization problem. The graphic shown in the second part of the figure demonstrates that this method reduces the search space in the second phase by an order of magnitude or even more. It is also observed that the first two variables (buffer places in the Petri Net model) are not so sensible to changes in the profit value while the other two variables (machine and model change variables) undergo a greater change in the profit value with their change.

Figure 2 shows similar results to the ones obtained for the first example. This second one shows also that the reduction has good behavior but not as good as in the first one.

3.2 Parameter Temperature Reduction Strategy

The idea behind the second scheme is based on the search space regions that have been visited during the first phase. It tries to extract information from the path that the optimizer took during the first phase.

With this information, the second phase is accelerated by reducing the temperature parameters of the optimizer. These temperatures are chosen in relation to the coefficient of variation (CV) of every variable included in the optimization problem. Because the coefficient indicates how disperse the data are, a low CV for a variable means that this variable can have a lower temperature to concentrate more on smaller region in the second phase. Due to this argument the CV is chosen independently for every variable. The reduction is in this case in the number of simulations computed due to the acceleration in the ASA optimization process.

In a first heuristic a linear relation between the CV and the temperature of the corresponding parameter is considered. A second variant applies a quadratic function. In the two cases under study the first option results in higher temperatures if the constant that multiplies the CV is the same.

Finally, a combination of the two strategies (search space and temperature reduction) is considered. The results for the two considered manufacturing system examples show that while keeping the original solution quality, the computational effort can be reduced by 50 percent in addition to the previously developed techniques.

Figures 3 and 4 show the results obtained for the two examples applying the new technique to the first phase results to obtain the new temperature scheme for the parameters in the second optimization phase. Later, we present the results corresponding to the application of this second phase and the time gain.

The results of Figures 3 and 4 have been obtained considering all the solutions of the first phase. It is however possible to restrict that set by only selecting the better points in a similar manner as previously done in the search space reduction strategy (c.f. Section 3.1). The reader can check that the original temperature values are decreased from the default value, 1. This decreases the possibility of jumping far from the actual region. The lower the parameter temperature value is, the lower the probability of jumping far from the original point gets.

	Parameter 1		Parameter 2		Parameter 3		Parameter 4	
	Min	*Max*	*Min*	*Max*	*Min*	*Max*	*Min*	*Max*
In. (100%)	1	4	2	30	0.20	4.00	0.20	4.00
50%	1	4	3	30	0.35	3.97	0.20	4.00
30%	1	4	4	30	0.57	3.97	0.20	4.00
20%	1	4	4	30	0.91	3.97	0.20	4.00
10%	1	4	5	30	1.90	3.95	0.20	0.92
5%	1	2	5	21	2.50	3.90	0.20	0.55

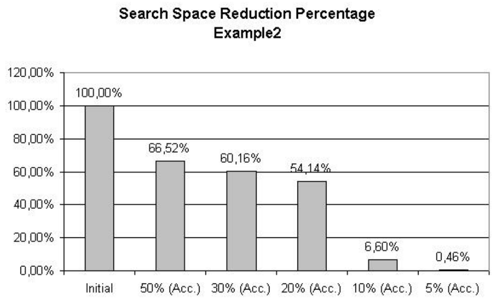

Figure 2: Search Space Reduction Strategy and Gain Percentage for the FMS in Example 2 in [12]

	CVPar1	TempPar1	CVPar2	TempPar2	CVPar3	TempPar3	CVPar4	TempPar4
Initial	—	1	—	1	—	1	—	1
Temp	0.81	0.2862	0.59	0.1953	1.07	0.3780	0.32	0.2253
TempQuad	0.81	0.0819	0.59	0.0382	1.07	0.1429	0.32	0.0507

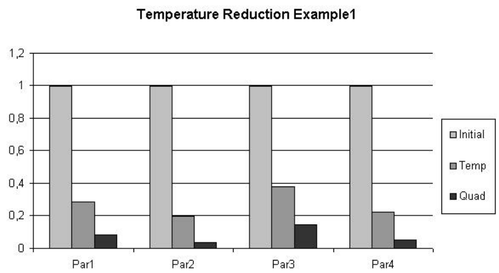

Figure 3: Variation Coefficient and Temperature for Example 1

	CVPar1	TempPar1	CVPar2	TempPar2	CVPar3	TempPar3	CVPar4	TempPar4
Initial	—	1	—	1	—	1	—	1
Temp	0.45	0.26	0.66	0.24	0.19	0.03	1.09	0.11
TempQuad	0.45	0.0678	0.66	0.0619	0.19	0.0013	1.09	0.0126

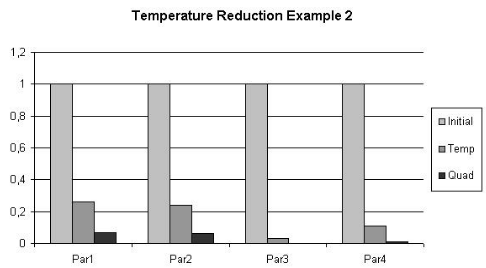

Figure 4: Variation Coefficient and Temperature for Example 2

4 Second Approach

The second approach presented uses a parameterized simulation process to decrease the computational effort of the optimization of any stochastic Petri net. The simulation parameter that is controlled for this variable simulation scheme is the relative error percentage of the simulation process. The stopping condition of each simulation run depends on the percentage of error. The error percentage parameter is changed according to the optimization algorithm process. The optimization parameter that guides this simulation parameter change is the temperature of the Simulated Annealing algorithm. The idea behind that approach is that during an early stage of the optimization the profit function value is still changing a lot, and the exact value of it is much less important than during the final phase. In the simulated annealing algorithm, the decreasing temperature is related to how close to the final area the current parameter already is. Different possible equations that relate the simulation parameter and the temperature of the simulated annealing algorithm have been considered.

In order to check the possible gain obtained in the optimization process using this variable simulation method, different values for the simulation parameters are considered and applied to the two application examples. Simulations for these values are run to check the possible efficiency of this method.

Figure 5 presents results for the two examples and the values of the relative error ranging between 30 and 1 percent. The first three columns of the table corresponds to the three different models that can be chosen for Example 1 (one of the optimization variables is the policy/model applied) while the fourth column correspond to the second example results. The first two rows include values ranging between 30-10 and 9-5 because the computational time results were similar for this interval of relative error values. This table shows the gain in computational time using this new technique. The reader can observe that the more accurate (lower values of the relative error) we are the more time consuming the tasks are. As expected, the simulation time computed in this case is clearly increasing as we decrease the relative error values, in some cases even by two orders of magnitude. These results also show that due to the relation established between the simulation parameter and the temperature parameter, in the beginning of the optimization process the simulations are really fast and not completely accurate while in the final part of the optimization process the simulations are slower and more accurate as it was intended.

5 Results

Now some computational results and profit quality values are presented to compare previous works with the heuristics explained in this paper.

Table 1 presents the results for all the experiments considered here. Columns 2 and 6 (Comp. Time) show the time gain compared with the basic (two phases approach, in second row). Positive values of these columns mean time gain percentage (with respect to the old values) while negative values mean worse time. Also columns 3 and 7 show the time effort expressed in hours, minutes and seconds. Columns 5 and 9 (Profit Value) show the best profit obtained with the different experiments, while columns 4 and 8 (Result Error) show the profit loss compared with the old previous approaches. Positive values of this column mean loss in the quality of the solution while negative values mean a gain in the quality of the solution.

The computational effort gain results show the generalized decrease in the computational effort except in the case of search space reduction strategy method for the first example, where the time spent during the optimization process increases. For the rest of the experiments the computational gain is substantial. The quality in the solutions obtained is, in most of the experiments, good enough to show the advantages of the methods proposed here. There are only two experiments where the solution is extremely worse than the best solution obtained.

The results for the second example are clearly better than for the first one. Here, 13 of the experiments are good while the remaining three are reasonably good. In this example no experiment is having a higher computational effort than the original two phase method.

6 Conclusions

The use of a combined two phase strategy has been proven as a good computational improvement for the optimization. The effort can be reduced using more information from the first phase. The Space Reduction and Temperature Reduction Strategy reduce this computational effort while keeping quality of results. Another method, based on variable simulation runs, is presented for Petri Net models that do not fulfill the conditions for the two phase optimization method.

We have checked the validity of the new methods with two examples from the area of manufacturing systems. The main advantage of these methods is that most of the experiments checked have reduced the time effort. The search space reduction strategy has improved the quality of the solution except in one case while the Temperature reduction strategy improves the effort always and the variable simulation approach improves drastically also the effort without loosing excessively quality.

References

[1] E. Aarts, J. Korst. "Simulated Annealing and Bolzmann Machines". *Wiley* (1989)

[2] G. Balbo, M. Silva (eds.). "Performance Models for Discrete Events Systems with Synchronizations: Formalism and Analysis Techniques" *MATCH Summer School, Jaca, September* (1998)

Rel. Err. Value	Ex1Push	Ex1Kan	Ex1Dem	Example2
30-10 Percent	1 Min. 20 Sec.	4 Sec.	4 Sec.	11 sec
9-5 Percent	8 Min. 40 Sec.	4 Sec.	4 Sec.	12 Sec.
4 Percent	11 Min. 50 Sec	5 Sec.	5 Sec.	22 Sec.
3 Percent	11 Min. 55 Sec.	16 Sec.	7 Sec.	22 Sec.
2 Percent	2 H. 15 Min.	24 Sec.	16 Sec.	48 Sec.
1 Percent	5 h. 4 Min.	1 Min. 5 Sec.	1 Min. 13 Sec.	1 Min. 44 Sec.

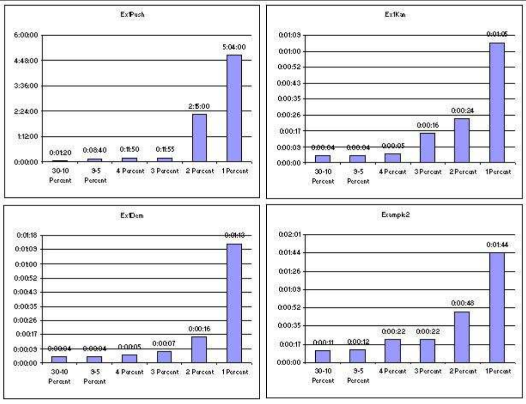

Figure 5: Simulation times with Relative Error Variation

Experiment	Example 1				Example 2				
Result	Time Gain	Comp. Time	Result Error (%)	Profit Value	Time Gain	Comp. Time	Result Error (%)	Profit Value	
One Phase	-334,15,00	11:15:00	0,00	8442,73	-305'97	38:24:00	0,00	5339,00	
Two Phase	0,00	3:22:00	0,00	8442,73	0,00	12:33:00	0,00	5339,41	
SSR	50%	-80,69	6:05:00	-1,27	8549,66	17,80	10:19:00	2,59	5200,87
	30%	-80,69	6:05:00	-1,27	8549,66	15,80	10:34:00	3,62	5146,13
	20%	-80,69	6:05:00	-1,27	8549,66	20,19	10:01:00	3,05	5176,57
	10%	-129,21	7:43:00	-1,26	8548,86	16,60	10:28:00	-10,45	5897,13
	5%	82,67	0:35:00	89,96	847,88	66,00	4:16:00	-12,59	6011,48
TRS	Linear (100%)	20,79	2:40:00	0,36	8412,75	57,77	5:18:00	-5,74	5645,92
	Linear (20%)	0,50	3:21:00	-1,26	8548,86	47,68	6:34:00	-5,12	5612,92
	Linear (10%)	0,99	3:20:00	-1,26	8548,86	42,63	7:12:00	-2,86	5492,17
	Linear (5%)	83,66	0:33:00	89,45	890,89	39,44	7:36:00	-2,76	5486,68
TRS	Quadratic (100%)	39,11	2:03:00	0,36	8412,75	84,86	1:54:00	7,33	4947,92
	Quadratic (20%)	47,52	1:46:00	-2,43	8647,90	66,93	4:09:00	7,34	4947,56
	Quadratic (10%)	47,52	1:46:00	-2,43	8647,90	66,80	4:10:00	-5,24	5619,34
	Quadratic (5%)	83,66	0:33:00	90,07	838,50	61,09	4:53:00	5,98	5020,30
VS	Linear Relerr	98,68	0:03:17	1,13	8347,16	77,41	5:18:00	-10,71	5645,92
	Quadratic Relerr	98,68	0:03:17	1,13	8347,16	77,41	5:18:00	-10,71	5645,92
	Root Relerr	98,68	0:03:17	1,13	8347,16	77,41	5:18:00	-10,71	5645,92

Table 1: Computational Effort Gain and Profit Loss for Examples

[3] J. Campos, G. Chiola, M. Silva. "Properties and Performance Bounds for closed Free Choice synchronized Monoclass Networks". *IEEE Transactions on Aut. Control Vol. 36 Num. 12 pp. 1368-1382.* (1991)

[4] J. Campos and M. Silva. "Structural Techniques and Performance Bounds of Stochastic Petri Net models". *Advances in Petri Nets 1992, Vol. 609 of LNCS. Springer Verlag, pp. 352–391.*

[5] F. DiCesare, G. Harhalakis, J.M. Proth, M. Silva, F.B. Vernadat. "Practice of Petri Nets in Manufacturing". *Chapman-Hall* (1993)

[6] M. Gen and R. Cheng. "Genetic Algorithms and Engineering Design". *Wyley* (1997)

[7] L. Ingber. "Adaptive simulated annealing (ASA): Lessons learned". *Journal of Control and Cybernetics Vol. 25 No. 1 pp. 33–54* (1996)

[8] C.L. Reeves. "Modern Heuristic techniques for Combinatorial Problems". *Wiley* (1993)

[9] A. Zimmermann, J. Freiheit, R. German, and G. Hommel. "Petri Net Modelling and Performability Evaluation with TimeNET 3.0." *TOOLS'2000, LNCS 1786, pp. 188-202, Illinois, USA* (2000)

[10] A. Zimmermann, D. Rodriguez, M. Silva. "Modelling and Optimization of Manufacturing Systems: Petri Nets and Simulated Annealing". *European Control Conference, Karlsruhe* (1999)

[11] A. Zimmermann, D. Rodriguez, M. Silva. "A Two Phase Strategy for DEDS: Application to a Manufacturing System". *Workshop on Discrete Event Systems, Ghent, pp. 291-298* (2000)

[12] A. Zimmermann, D. Rodriguez, M. Silva. "A Two Phase Optimization Method for Petri Net Models of Manufacturing Systems". *Journal of Intelligent Manufacturing, Vol. 12 No. 5 pp. 421-432* (2001)