SIMULATION AND MATHEMATICAL PROGRAMMING FOR A MULTI-OBJECTIVE CONFIGURATION PROBLEM IN A HYBRID FLOW SHOP

Pierpaolo Caricato
Antonio Grieco
Francesco Nucci

Department of Innovation Engineering
University of Lecce
Via per Monteroni
Lecce, 73100, ITALY

ABSTRACT

This paper introduces an application of simulation-based multi-objective optimization to solve a system configuration problem in a hybrid flow shop system. The test case is provided by a firm that manufactures mechanical parts for the automotive sector. We present an architecture that uses both discrete-event simulation and mathematical programming tools in order to solve the problem. The multiple-objective nature of the problem is preserved throughout the proposed approach, using Pareto-dominance concepts both to eliminate inefficient solutions within the proposed solution algorithm and to provide the user with efficient solutions. Mathematical programming is used to cull the required number of simulation runs. Computational results obtained using a real-world case study are reported. The proposed approach is benchmarked against a general purpose simulation-optimization engine in order to prove its effectiveness.

1 INTRODUCTION

In modern manufacturing systems, *production line*, one of the most efficient and adopted production layouts, has been improved with the introduction of flexible machines at some or all of its stages. This has led to a great enhancement in system's flexibility, but it has also led to a greater complexity in its management. The following issues can contribute to the complexity of the system's management: flexible machines needing a human supervision, intermediate buffers for unfinished parts, quality control stages and the introduction of loops in the workflow. Regardless of the specific system features, one of the most common objectives to be pursued when managing the system is the optimization in the use of the available resources, both in terms of machinery and human resources. The complexity of modern systems has often suggested the use of hybrid approaches, using specific features coming from management tools tra-

ditionally used as stand-alone solutions. Simulation and mathematical programming are among the methods that are commonly used for this purpose.

We address a configuration problem that arises in a firm that manufactures mechanical parts for the automotive industry. The production system can be classified as a *hybrid flow shop with variable processing times and finite capacity buffers*. Flexible machines are present at each stage, along with buffers for unfinished parts located before each stage. Several part types can be processed by the system: each buffer is composed of distinct buffers, one for each part type. The number of items to be processed for each part type is known. Each stage can adopt several configurations, each requiring the supervision of one or more workers, and its performance depend on the chosen configuration. The objective is to choose a system configuration pursuing two often contrasting objectives: the minimization of the required workers and the minimization of the makespan, i.e. the total time needed to process all the required items.

We propose a hybrid approach that uses simulation and mathematical programming integrated within an architecture that allows a coordinated use of the main features of both approaches, providing the system's manager with a set of efficient configurations to choose among. In the next section we present reference works that allow the collocation of our work in the existing literature. In the following section 3, the proposed method is presented. Aspects related to mathematical programming are analyzed in section 4, while the simulation model is presented in section 5. In section 6 we apply the proposed approach to a real-world case study and we benchmark it against a general purpose simulation-optimization engine. Finally, section 7 summarizes the paper results.

2 RELATED WORK

Multi-objective optimization is an interesting research area in the field of optimization methodologies, especially using

Evolutionary Algorithms (Deb 2001). However, state-of-the-art publications report relatively few attempts in the area of simulation-based multi-objective optimization. Special cases, in such a field, can be found in (Eskandari et al. 2005) and (Baesler and Sepúlveda 2001).

Evans et al. (1991) suggest a framework for the multicriteria optimization of simulation models by, first, discussing the unique difficulties of this problem area along with important problem characteristics, and then discussing the way that these problem characteristics would affect the choice of a particular technique.

Gupta and Sivakumar (2002) present a simulation-based multi-objective optimization method based on compromise programming for operation scheduling in semi-conductor manufacturing. The proposed method is used to solve a NP-hard problem that consists in scheduling a number of independent jobs on a single machine and finding a Pareto optimal solution. The objectives observed in the optimization strategy are average tardiness, cycle time, and machine utilization. A campaign of theoretical job-shop experiments is carried out using the proposed method.

A simulation optimization problem is an optimization problem where the objective function, constraints, or both are responses that can only be evaluated by computer simulation. An analytical expression of the objective function or the constraints does not exist. Computer simulation programs require much longer processing times than the evaluation of analytical functions. This makes the efficiency of the optimization algorithms more crucial. The most common formulation for optimization of systems through simulation has been the maximization or minimization of the expected value of the objective function of the problem.

In (Banks et al. 2000) various approaches are categorized according to the characteristics of the adopted algorithms. The main techniques that have been used can be divided into the following main categories:

- statistical procedures: sequential response surface methodology, ranking & selection procedures, and multiple comparison procedures;
- meta-heuristics: methods directly adopted from deterministic optimization search strategies, such as simulated annealing, tabu search, and genetic algorithms;
- stochastic optimization: random search, stochastic approximation;
- others, including ordinal optimization and sample path optimization.

An alternative technique classification, for single objective problems, presented in (Azadivar 1992) is:

- gradient based search
- stochastic approximation

- response surface
- heuristic search

When considering multi-objective problems, the concept of Pareto-optimality plays an utmost role in multi-objective problems: a vector $x^* \in S$ is said to be Pareto optimal for the considered problem if all other vectors $x \in S$ have a higher value for at least one of the objective functions $f_i(\cdot)$, or else have the same value for all objectives. S is the feasible set and $f_i, i \in \{1, \ldots, I\}$ with $I \geq 2$ is the array of objective functions, all to be minimized. Pareto optimal points are also known as efficient, non-dominated, or non-inferior points. An up-to-date review of Pareto multi-objective optimization can be found in Ehrgott (2005).

3 THE PROPOSED METHOD

The problem to be addressed has a double-faced objective: to minimize the number of workers required by the adopted system configuration (objective O1) and to minimize the blocking or starvation times within the system (objective O2). Given the highly different nature of such objectives, these aspects cannot be effectively translated into a single, numerical objective, such as a cost function. Hence, the problem has to be treated explicitly addressing its multi-objective nature.

3.1 Multiple Objectives

In the addressed problem, the values that can be assumed by one of the two pursued objectives can only be integer values: namely, the number of workers required by the adopted system configuration. Furthermore, in real world cases, a rough estimate of a reasonable range for this value can also be obtained. Hence, in the proposed approach, we can fix a range $\{\eta_{min}, \ldots, \eta_{max}\}$ of values for the O1 objective and iteratively address the problem obtained introducing the additional constraint to use a specified number $\overline{\eta}$ of workers iteratively chosen within the mentioned range, pursuing the O2 objective.

3.2 Simulation vs. Optimization

Theoretically, at each iteration, it would be possible to enumerate all the configurations requiring a fixed number of workers and then run a simulation to evaluate the configuration that achieves the best performance according to the O2 objective. Indeed, an approach entirely based on simulation techniques is not viable for the analyzed case, since it leads to a number of simulation runs needed that grows exponentially with the number of stages and the number of product types. This aspect will be deeply analyzed with reference to the case study in section 6.

On the other hand, an approach entirely based on a comprehensive mathematical model cannot be effectively adopted. The blocking and starvation times in the system are determined by a wide range of aspects: the quantities to be produced for each part type, the inventory state among inter-stage buffers, the processing rates at each stage, which depends both on the stage configuration and on the part type to be processed. Finally, the idle times at any stage can depend on the configurations chosen for all the preceding and following stages. Hence, a model including all these aspect, and in particular the last one, would grow exponentially in complexity with the number of stages to be considered.

3.3 Problem Decomposition

The problem can be effectively modeled through a mathematical model accepting a simplification of the system. In particular, we adopted a problem decomposition method similar to the one presented in Colledani et al. (2005). The system is considered in its whole complexity, but the interactions among stages are only considered couple-wise: i.e. only the idle times directly generated by each couple of consecutive stages are considered. This approach neglects the domino effect that can cause, for instance, a blocking phenomenon at the first stage to propagate through all the stages of the system but, as illustrated in sections 4 and 6, this rarely affects the results obtained when idle times are being couple-wise minimized. The details of the mathematical model are reported in section 4. It is important to note that such a mathematical model, even if solved to optimality for a fixed number of workers, cannot guarantee the actual optimality of the found solution in the real system, due to the introduced simplification.

3.4 Simulation Runs Reduction

The purpose of using such simplified mathematical model is to reduce the number of system's configurations to be studied through simulation. Only the best solutions for each fixed number of workers are, indeed, saved and will be used as an input for the simulation engine. The number of solutions to be considered is a parameter of the method. Figure 1 shows an example that better explains how simulation runs can be reduced with the proposed approach. The horizontal axis reports the number of workers needed by the system configuration, while the objective function of the mathematical model is reported on the vertical axis.

The values on the horizontal axis represent different runs of the mathematical model with the constraint to use exactly $\overline{\eta} \in \{\eta_{min}, \ldots, \eta_{max}\}$ workers in the system. The bold dots on the vertical line relative to $\overline{\eta}$ represent different system configurations, each requiring exactly $\overline{\eta}$ workers and characterized by the objective function value corresponding to the vertical coordinate of the dot. The best configuration
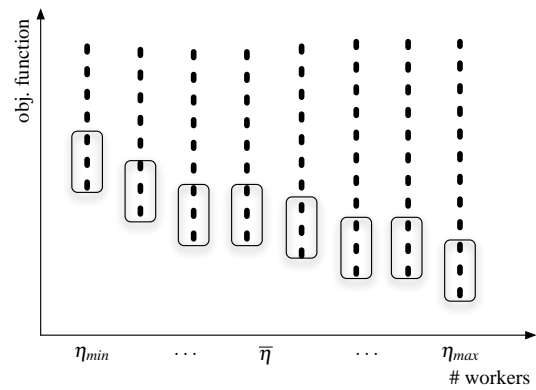

Figure 1: Configurations to be simulated.

for each $\overline{\eta}$ is represented by the lowest dot. However, since the mathematical model represents a simplified system, it is not guaranteed that the configuration corresponding to the lowest dot is actually the best configuration requiring $\overline{\eta}$ workers in the real system. Therefore, a number of good configurations are chosen to be tested using simulation. Such configurations are the ones contained in the rectangles in Figure 1: in the example, the three lowest – and hence best according to the mathematical model – configurations are chosen for each $\overline{\eta}$.

It is worth noting that the number of configurations for each $\overline{\eta}$ grows exponentially with the number of stages and that an exhaustive simulation of all possible system configurations is extremely time-consuming and rapidly becomes impossible for real world instances of the problem. Such consideration is confirmed by the analysis reported in section 6.

3.5 Method Overview

Figure 2 shows a graphical representation of the proposed approach. For each $\overline{\eta} \in \{\eta_{min}, \ldots, \eta_{max}\}$ the mathematical model described in section 4 is solved in order to store the h best solutions, where h is a user-defined number. The result can be shown as in Figure 1: the value of h determines the height of the selection rectangles. A further selection among the selected solutions is conducted using the concept of Pareto-optimality: if a solution achieves the same or better objective function than another one requiring less workers, than the solution with the higher $\overline{\eta}$ is not considered. The simulation process described in section 5 is then used to evaluate the actual performance parameters for the selected configurations. Similarly to the selection conducted before, only Pareto-optimal solutions are then reported to the decision maker.
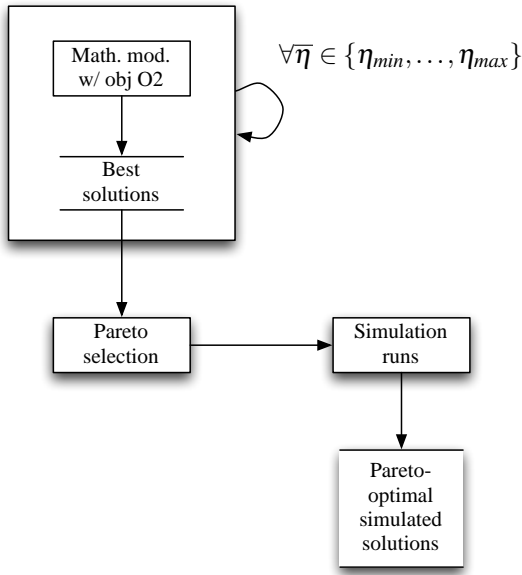
$$\forall \overline{\eta} \in \{\eta_{min}, \ldots, \eta_{max}\}$$

Figure 2: Method overview.

4 MATHEMATICAL PROGRAMMING

A mathematical programming approach is used, as described in section 3, in order to find the best configurations for the system once the number of workers to be used is fixed. The objective to be pursued is to minimize the blocking or starvation times within the system, while the constraints to be considered are relative to the number of workers to be used and to the configurations available for each stage and to the quantities of different part types to be manufactured.

The following notation is used throughout the paper:

P is the set of different part types processed by the system;

$p \in P$ is the index used to indicate a specific part type;

S is the set of stages available in the system;

$i \in S$ is the index used to indicate a specific stage;

C_{ip} is the capacity of the i-th buffer for p part type items;

W_{ip} is the number of partially processed p part type items available in the i-th buffer;

Q_p is the required number of p part type items.

The binary decision variables x_{ij} are defined in order to indicate whether the i-th stage uses the j-th configuration

or not. Variable x_{ij} is defined for $i \in S$ and $j \in \Gamma$, where Γ is the set of possible system's stages configurations. Each stage i is characterized by a set of possible configurations $\Gamma_i \subset \Gamma$. The whole set Γ can be hence partitioned as follows: $\Gamma = (\Gamma_1 | \ldots | \Gamma_S)$ and $x_{ij} = 0$ $\forall i \notin \Gamma_i$. Hence, for each configuration:

$j \in \Gamma$ is the index used to identify a specific configuration;

n_j is the number of workers required by the j-th configuration;

r_{jp} is the processing time of a part type p item when the j-th configuration is used.

Finally, the allocation of workers along the stages is defined by:

$$\alpha = (n_{j_1}, \ldots, n_{j_{|S|}}) \quad j_i \in \Gamma_i \quad \forall i \in S$$

In other words, α indicates the number of workers per each stage.

An example of the adopted notation is given in Figure 3. Three stages and two part types are reported. For each stage, one or two workers can be used. Two worker allocations A and B are represented. The first one $\alpha_A = (2, 1, 1)$ adopts two workers for stage 1 (configuration j_2), one worker for the remaining stages (configurations j_3 and j_5). The second $\alpha_B = (2, 2, 1)$ uses an additional worker in stage 2 (configuration j_4)

Given the fixed number of workers $\overline{\eta}$ to be used by the overall system configuration, the following constraint ensures that such condition is verified:

$$\sum_{i \in S} \sum_{j \in \Gamma_i} n_j x_{ij} = \overline{\eta}$$

For each stage, exactly one configuration has to be chosen, hence:

$$\sum_{j \in J_i} x_{ij} = 1 \quad \forall i \in S$$

and each configuration can be assigned to at most one stage, as guaranteed by the following constraint:

$$\sum_{i \in S} x_{ij} \leq 1 \quad \forall j \in J$$

A possible way of achieving the objective to minimize the blocking or starvation times within the system is to minimize the average makespan calculated for each couple of consecutive stages, in accordance with the decomposition
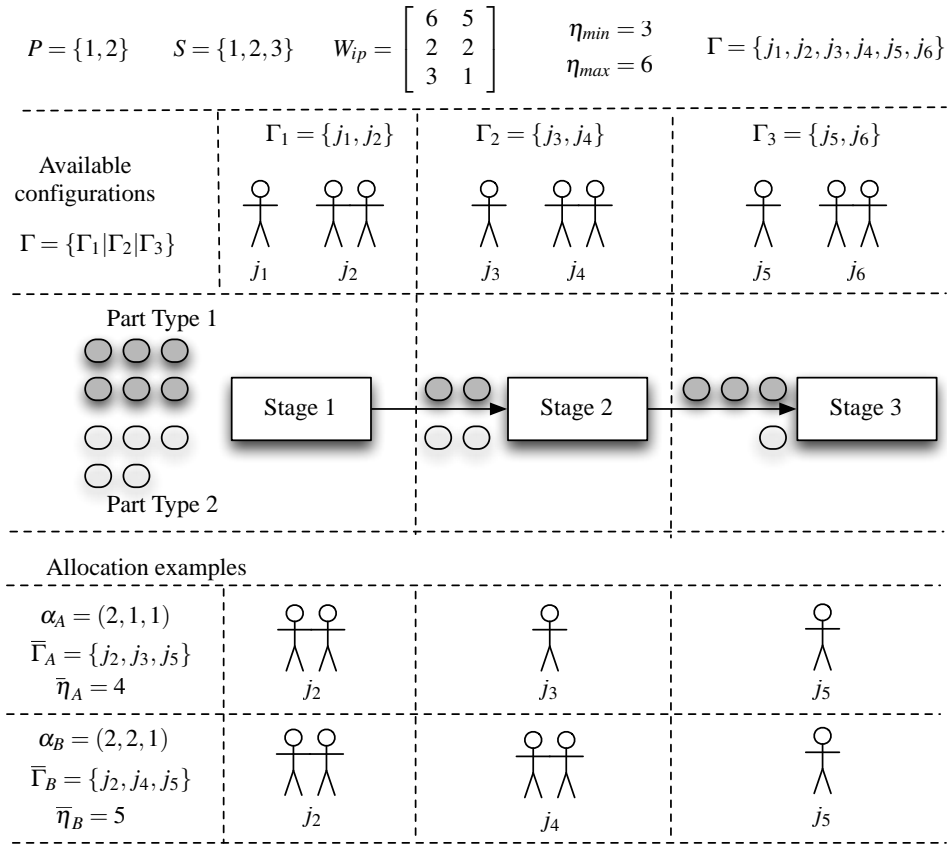
Figure 3: Example of the adopted notation.

principle introduced in 3.3. The makespan for the generic couple of consecutive stages $(i, i+1)$ on part type p when stage i adopts configuration j and stage $i+1$ uses configuration k is denoted by m_{ijkp}. Thus, the overall objective function to be minimized is written as follows:

$$\min \sum_{p \in P} \sum_{i=1}^{|S|-1} \sum_{j \in \Gamma_i} \sum_{k \in \Gamma_{i+1}} m_{ijkp} x_{ij} x_{(i+1)k}$$

The problem can be considered as a variant of the well known QAP (Quadratic Assignment Problem). In general, the QAP consists in the assignment of facilities to location, given a material flows matrix that contains the expected flows between couples of facilities, and a distance matrix, that geometrically characterizes the distances between couples of locations. The general objective is to minimize the overall assignment cost, calculated as the sum of the distance-flow products corresponding to the selected assignment.

The classic QAP problem was introduced in Koopmans and Beckmann (1957), while an up to date review of QAP related literature can be found in Loiola et al. (2007). In

general QAP, the number of facilities exactly matches the number of locations. The stages in the considered system can be modeled as locations, while configurations can be thought of as facilities to be assigned to each location/stage. The assignment cost is given by the makespan associated with each configuration chosen for each stage. Unlike classic QAP, the number of configurations to be assigned is not equal to the number of stages. The problem has been solved using ILOG CPLEX^{TM}, a commercial MIQP (Mixed Integer Quadratic Problem) solver.

5 SIMULATION MODEL

A deterministic discrete event simulation model has been adopted in order to evaluate the actual system performance for a specific configuration $\overline{\Gamma} = \{j : x_{ij} = 1\}$. Such a model has been developed with the Arena Rockwell^{TM} software ver. 7. The input data set for the simulation model consists in:

- $j \in \overline{\Gamma}$
- $W_{ip}, C_{ip} \quad \forall i \in S, \quad \forall p \in P$

- $Q_p \quad \forall p \in P$
- $r_{jp} \quad \forall j \in \overline{\Gamma} \quad \forall p \in P$

The entities represented in the simulation model are the parts to be produced. For each buffer stage, a priority based rule is adopted to determine the next part to be processed. At a given time t, the part type having the highest value of the priority index $\phi_p(t)$ is selected, see (1). Such an index is based on the number of finished items for a given part type $f_p(t)$.

$$\phi_p(t) = 1 - \frac{f_p(t)}{Q_p} \quad \forall p \in P \qquad (1)$$

At each stage, the part type having the highest priority index is processed first. Note that, at a given stage i, if the input part type buffer is full, the previous stage $i-1$ does not process such part type in order to avoid the blocking phenomenon.

Simulation model events are the start/end of a part processing for a given stage. The simulation stops when all the required items Q_p are produced for each part type. Finally, the simulation outputs are the system makespan and the average usage rate for each resource stage.

6 CASE STUDY

The validation of the proposed approach has been conducted with an experimental campaign performed on a real world case study in the automotive sector.

6.1 EXPERIMENTAL CAMPAIGN

The configuration problem consists in the allocation of workers on an in-line production system with a fixed number of stages and part types. The objective consists in finding the configuration minimizing the makespan and worker number. The experimental campaign consists in five different production scenarios reported in Table 1. As it is possible to note, the number of stages and part types varies from 5 to 10. For each stage, the number of workers that can be used varies from 1 to 3.

Table 1: Experimental campaign scenarios.

Id Scenario	No. stages $|S|$	No. part types $|P|$
1	5	8
2	6	6
3	8	5
4	8	8
5	10	10

In the following, the case study is synthetically described on the basis of probability functions inferred from an analysis of the real production system.

For each part type, the number of parts to be produced Q_p has been generated from a uniform probability distribution with min/max values equal to 60 and 120 ($Q_p : \lfloor UNIF(60,120) \rfloor$ in the following). For each stage and part type, the number of parts waiting to be processed W_{ip} is defined in (2). Note that parts waiting to be processed at stage 1 are raw parts. The buffer capacity is $C_{ip} : \lfloor UNIF(8,60) \rfloor$. The probability that a part type is not processed at a given stage is 0.3. If a part type is processed at a given stage, the processing time values r_{jp} are inferred from the probability density functions reported in (3). Different probability density functions are used in order to achieve the result of having larger processing times when less workers are used.

$$W_{ip} : \begin{cases} \lfloor UNIF(2,20) \rfloor & i > 1 \\ \lfloor UNIF(30,70) \rfloor & i = 1 \end{cases} \qquad (2)$$

$$r_{jp} : \begin{cases} UNIF(20,100) & \text{mins}, \quad n_j = 1 \\ UNIF(12,60) & \text{mins}, \quad n_j = 2 \\ UNIF(8,40) & \text{mins}, \quad n_j = 3 \end{cases} \qquad (3)$$

The experimental campaign is used to compare the performance of the proposed method with another tool to configurate the system. On one hand, if the number of feasible configurations is limited, an exhaustive approach can be adopted by simulating any configuration Γ (see Table 2). On the other, a simulation optimization tool has to be adopted. We selected OptQuest[TM], included in the Arena Rockwell package, in order to benchmark our methodology. OptQuest is an optimization tool customized for analyzing the results of simulation runs conducted in Arena. OptQuest includes sampling techniques and error control to find better answers, and incorporates algorithms based on tabu search, scatter search, integer programming, and neural networks. A time limit of two hours has been fixed in order to let OptQuest find the best solution (see Table 2).

In the considered case study, the variables controlled by OptQuest are the number of workers n_j allocated on each stage for a given configuration j. A constraint is introduced in order to have a fixed number of workers $\overline{\eta}$ in the overall system. The performance parameter is the makespan to be minimized. As reported in section 3.5, an analysis is performed for any $\overline{\eta} \in \{\eta_{min} \dots \eta_{max}\}$. Since the number of workers that can be allocated to each stage varies between 1 and 3, then $\eta_{min} = |S|$ and $\eta_{max} = 3|S|$. Then, the Pareto analysis is performed and the final solution list is issued.

Table 2: Analysis of computer processing times with different approaches.

Id	Computational time		
Scenario	Exhaustive approach	Proposed approach	OptQuest approach
1	4h 3'	5'	16'
2	12h 9'	20'	55'
3	4d 22h 27'	33'	2h*
4	5d 16h 41'	38'	2h*
5	68d 8h 15'	55'	2h*

<div align="right">* time limit</div>

6.2 COMPUTATIONAL RESULTS

For each scenario, the benchmark between the proposed approach and OptQuest is performed. The Pareto optimal solution set is reported together with the corresponding makespan using both the proposed approaches (see Table 3, 4, 5, 6 and 8) and OptQuest (see Table 3, 4, 5, 7 and 9). For scenario 1, 2 and 3 both approaches lead to the same solution set because of the limited number of available configurations. Instead, for scenario 4 and 5 different results are obtained. In Figure 4 and 5, a graphical comparison can be performed between the two approaches. As it is possible to note the solution set obtained with the proposed approach dominates the OptQuest set. Moreover, the proposed approach obtains such results in less time than OptQuest. Such an outcome is obtained because of the ability of our optimization model to predict the most suitable solutions to be investigated with the simulation analysis. Instead, the simulation optimization approach performed with OptQuest does not take into account the internal system description and executes a sequence of simulation considering only the control variables and performance parameters.

Table 3: Pareto optimal solutions for scenario 1 using both the proposed approach and OptQuest.

No. of workers	Worker allocation α	Makespan (min)
5	(1,1,1,1,1)	34518
9	(1,2,2,2,2)	24291
10	(2,2,2,2,2)	18512
14	(2,3,3,3,3)	13009

7 CONCLUSIONS

In this work, we proposed a new hybrid technique to solve a system configuration problem in a manufacturing system. We addressed the problem using a mathematical program-

Table 4: Pareto optimal solutions for scenario 2 using both the proposed approach and OptQuest.

No. of workers	Worker allocation α	Makespan (min)
6	(1,1,1,1,1,1)	30526
8	(1,1,1,2,1,2)	23524
11	(1,2,2,2,2,2)	19914
12	(2,2,2,2,2,2)	16458
14	(2,2,2,3,2,3)	12721
18	(3,3,3,3,3,3)	11229

Table 5: Pareto optimal solutions for scenario 3 using both the proposed approach and OptQuest.

No. of workers	Worker allocation α	Makespan (min)
8	(1,1,1,1,1,1,1,1)	27297
9	(1,1,1,1,1,2,1,1)	18772
10	(1,1,1,1,1,2,2,1)	15821
15	(2,2,1,2,2,2,2,2)	14670
16	(2,2,1,2,2,3,2,2)	10219

ming approach integrated with a discrete event simulation tool. The mathematical problem performs a linear decomposition of the production system in order to reduce the set of solutions on which the simulation analysis is executed. The benchmark of the proposed approach has proven its effectiveness against a general purpose simulation-optimization tool.

REFERENCES

Azadivar, F. 1992. A tutorial on simulation optimization. In *WSC92*, 198–204.

Table 6: Pareto optimal solutions for scenario 4 using the proposed approach.

No. of workers	Worker allocation α	Makespan (min)
8	(1,1,1,1,1,1,1,1)	32264
9	(1,1,1,1,1,2,1,1)	28275
12	(1,1,1,2,2,2,2,1)	26237
13	(1,2,1,2,2,2,2,1)	20200
17	(2,2,2,2,2,3,2,2)	14670
21	(2,3,2,3,3,3,3,2)	11996

Table 7: Pareto optimal solutions for scenario 4 using OptQuest.

No. of workers	Worker allocation α	Makespan (min)
8	(1,1,1,1,1,1,1,1)	32264
12	(1,1,1,2,2,2,2,1)	26237
13	(1,2,1,2,2,2,2,1)	20200
18	(2,2,2,3,2,3,2,2)	14680
22	(2,3,2,3,3,3,3,3)	12298

Table 8: Pareto optimal solutions for scenario 5 using the proposed approach.

No. of workers	Worker allocation α	Makespan (min)
10	(1,1,1,1,1,1,1,1,1,1)	49177
11	(1,1,1,1,1,1,1,1,2,1)	32742
12	(1,1,1,1,1,2,1,1,2,1)	30494
15	(1,1,1,2,2,2,2,1,2,1)	25672
21	(1,2,2,2,2,3,2,2,3,2)	17036

Figure 4: Pareto optimal solutions for scenario 4.

Figure 5: Pareto optimal solutions for scenario 5.

Baesler, F., and J. Sepúlveda. 2001. Healthcare ii: multi-objective simulation optimization for a cancer treatment center. In *WSC '01: Proceedings of the 33nd conference on Winter simulation*, 1405–1411. Washington, DC, USA: IEEE Computer Society.

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete event systems simulation*. Englewood Cliffs, NJ: Prentice Hall.

Colledani, M., A. Matta, and T. Tolio. 2005. Performance evaluation of production lines with finite buffer capacity producing two different products. *OR Spectrum* 27 (2-3):243–263.

Table 9: Pareto optimal solutions for scenario 5 using OptQuest.

No. of workers	Worker allocation α	Makespan (min)
10	(1,1,1,1,1,1,1,1,1,1)	49177
11	(1,1,1,1,1,1,1,1,2,1)	32742
12	(1,1,1,1,1,2,1,1,2,1)	30494
14	(1,1,1,2,1,2,2,1,2,1)	28759
17	(1,1,2,2,2,2,2,2,2,1)	24844
20	(1,2,2,2,2,2,2,2,3,2)	19231
23	(1,2,2,3,3,3,2,3,3,2)	18318
24	(2,2,2,2,3,3,2,3,3,2)	17949
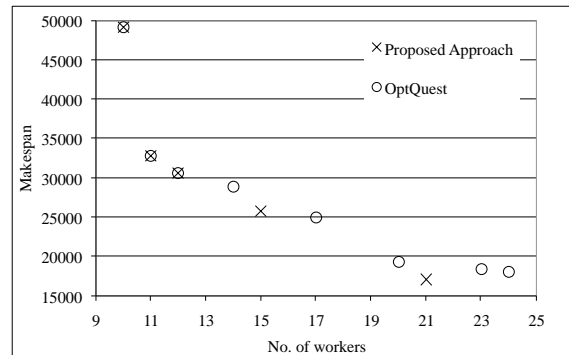
Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*. Wiley.

Ehrgott, M. 2005. *Multicriteria optimization*. Springer.

Eskandari, H., L. Rabelo, and M. Mollaghasemi. 2005. Multiobjective simulation optimization using an enhanced genetic algorithm. In *WSC '05: Proceedings of the 37th conference on Winter simulation*, 833–841: Winter Simulation Conference.

Evans, G. W., B. Stuckman, and M. Mollaghasemi. 1991. Multicriteria optimization of simulation models. In *WSC '91: Proceedings of the 23rd conference on Winter simulation*, 894–900. Washington, DC, USA: IEEE Computer Society.

Gupta, A., and A. Sivakumar. 2002. Semiconductor manufacturing: simulation based multiobjective schedule optimization in semiconductor manufacturing. In *WSC '02: Proceedings of the 34th conference on Winter simulation*, 1862–1870: Winter Simulation Conference.

Koopmans, T., and M. Beckmann. 1957. Assignment problems and the location of economic activities. *Econometrica* 25 (1):53–76.

Loiola, E., N. de Abreu, P. Boaventura-Netto, P. Hahn, and T. Querido. 2007. A survey for the quadratic assignment problem. *European Journal of Operational Research* 176 (2):657–690.

AUTHOR BIOGRAPHIES

PIERPAOLO CARICATO is member of the research staff in technology and manufacturing systems at the University of Lecce, Dep. of Innovation Engineering. His research interests are in exact and heuristic optimization methods for material-handling systems design and supply-chain planning. His email address for these proceedings is <pierpaolo.caricato@unile.it>.

ANTONIO D. GRIECO is an associate professor of technology and manufacturing systems in the University of Lecce, Dep. of Innovation Engineering. His research interests are in tool and workpiece management in flexible manufacturing systems plants, scheduling, discrete event simulation and fuzzy-sets theory. He's a member of the Associazione Italiana Tecnologia Meccanica (AITEM), the Italian Society of Computer Simulation (ISCS) and the Society for Modeling and Simulation (SCS). His email address for these proceedings is <antonio.grieco@unile.it>.

FRANCESCO NUCCI is assistant professor of technology and manufacturing systems in the University of Lecce, Dep. of Innovation Engineering. He is mainly interested in simulation under uncertain conditions and manufacturing system scheduling. He's a member of the Associazione Italiana Tecnologia Meccanica (AITEM). His email address for these proceedings is <francesco.nucci@unile.it>.