

Scalability Planning for Reconfigurable Manufacturing Systems

Wencai Wang and Yoram Koren

The University of Michigan, Ann Arbor, USA

Journal of Manufacturing Systems Vol. 31, No. 2, pp. 83-91, 2012

1. Introduction

Manufacturers today face more challenges than ever before due to the highly volatile market, which creates large fluctuations in product demand. To remain competitive, companies must design manufacturing systems that not only produce high-quality products at low cost, but also respond to market changes in an economical way [1].

Reconfigurable manufacturing systems (RMS) have been suggested by Koren et al [2] as a solution to address the needs for meeting the changing product demands. This has been recognized and supported later by other researchers [3-7]. From the viewpoint of RMS, a manufacturing system should be designed in such a way that it can be rapidly and cost-effectively reconfigured to the exact capacity needed to match the market demand. The capability of manufacturing systems to adapt their throughputs to changing demands is called scalability.

Scalability is an important system design characteristic in markets with volatile demand, and its cost-effective solution requires knowledge from engineering and business [8]. Researchers at the ERC/RMS [9], have addressed system scalability since the late 1990's [10], and issued a patent that deals with strategies to change production capacity in reconfigurable manufacturing systems (RMS) [11]. They developed one of the first algorithms that address capacity scalability [12], but this early algorithm was limited to upgrading the capacity of serial lines only. A more comprehensive approach was presented in [13] where scalability was analyzed as one of the critical issues in designing large, complex machining systems. Capacity scalability may be also achieved by scaling the capacity of individual pieces of equipment [5,6,14,15,16], but the most practical approach to system scalability adding machines to existing manufacturing systems, and in this cases the original system layout design is critical for achieving cost-effective scalability [17].

A dynamic model for capacity scalability analysis in reconfigurable manufacturing systems is introduced in [4]. This dynamic model is associated with minimizing the delay in scaling the system's capacity and thereby improving the RMS performance in response to sudden demand changes. However, in this current paper we deal with optimizing the original system layout [3,5] such that adding machines when needed by the market demand will be done quickly and cost effectively. Simultaneously with adding machines, also the material handling system must be adapted to serve the new added machines. There are cases in which several AGVs form the material transport system [18], but although AGVs facilitate the part transfer to and from the new machines, AGVs are expensive and slow, and therefore are not regarded as a cost-effective solution. There are cases in which RMS are designed to produce several products simultaneously [19, 20]. In such systems the capacity design issue is more complex and it is beyond the scope of this paper.

With the advancement of machine technologies over the past decade, the production of medium-to-high volume, large size mechanical parts, such as automotive powertrain components, has undergone a transformation. Dedicated transfer lines with dedicated machine stations are being replaced with systems composed of flexible CNC machine tools. As shown in Fig. 1, this system architecture is composed of multiple parallel CNC machines at each stage, with all machines performing exactly the same machining tasks, [21]. Such configurations of parallel identical machines in each stage, with material transfer between the stages (also known as crossover) improve throughput and reduce work-in-process inventories [22].

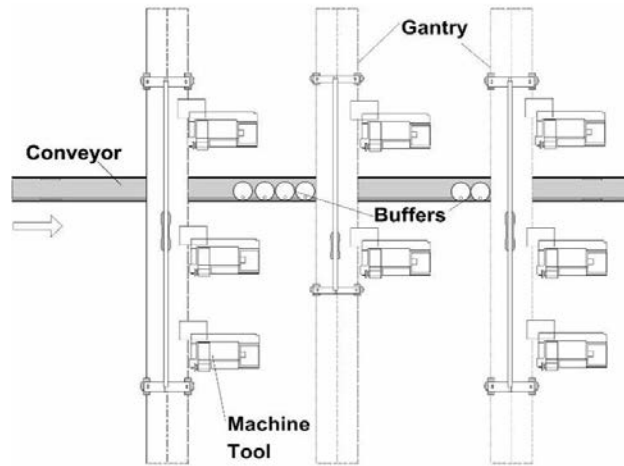


Figure 1: Schematic Layout of Production Lines with Conveyors and Gantries

Each manufacturing system is designed with a specific capacity in mind to fulfill a planned forecasted demand. However if the forecast for an annual product sale is between 250,000 and 300,000 units, marketing dictates building a capacity for 300,000 units. Therefore, even if a system is optimally designed, capacity may be still wasted when the real demand is significantly lower than the full planned capacity. When considering the entire life cycle of a manufacturing system the periods in which the system is operated at the full capacity are usually short [23]. If, however, the investment in the excess capacity (for 50,000 units in this example) could be delayed until it is actually needed, the system lifetime cost can be significantly reduced. A system design-for-scalability means that a manufacturing system is designed in a way that enables a rapid capacity upgrade to meet a larger demand, exactly when needed.

This paper introduces a practical design-for-scalability method for reconfigurable manufacturing systems comprised of reconfigurable and/or CNC machines tools. A scalability planning methodology is presented to determine the most economical way to add machines to an existing system to match a new market demand. It does so through concurrently changing system configuration and rebalancing the system. The remainder of this paper is structured as follows: Section 2 defines the system scalability and describes the concept of incrementally scaling system capacity. Section 3 introduces a mathematical formulation to minimize the total number of machines to be added by concurrently reconfiguring and rebalancing the system. Section 4 proposes heuristic algorithm based on a genetic algorithm. Section 5 presents case study to validate the proposed approach. Conclusions are presented in Section 6.

2. Defining System Scalability

To adapt the throughput of manufacturing systems to the fluctuations in product demand, the system capacities must be adjusted quickly and cost-effectively. Capacity scalability of manufacturing systems is a necessary characteristic needed for rapidly adjusting the production capacity in discrete steps, allowing thereby a given system's throughput to adjust from one yield to another to meet changing market demands. We define system scalability, in percentage, as:

$$\text{System Scalability} = 100 - \text{smallest incremental capacity in percentage}$$

If the minimal capacity increment by which the system output can be adjusted to meet new market demand is small, then the system is highly scalable. For example, if a serial line (Fig. 3a) needs to increase its production capacity to satisfy a larger market demand, an entire new line must be added. The step-size of this addition doubles the production capacity of the system. Mathematically, the minimum increment of adding production capacity in a serial line is 100% of the system, i.e., adding a whole new line, making the scalability of a serial line 0%. Doubling the line capacity will be expensive because there is no guarantee that the extra capacity will ever be fully utilized, risking a substantial financial loss. Thus, zero scalability means that in order to increase the system capacity, the entire production line must be duplicated.

Dedicated lines do not have scalable capacity and cannot cope with large fluctuations in product demand. This challenge can only be met by flexible or reconfigurable manufacturing systems which are composed of singular CNC machines, as these systems are scalable in small increments accomplished by adding individual machines can be added as a need arises.

Similar scalability calculations for the other systems in Fig. 3 show: Configuration b has a scalability of 50% and Configuration c has 67%. Configurations d and e have a scalability of 84%; the highest possible for 6-machine configurations. A minimum increment of only one sixth of the system (16%) —in these cases, one machine— can be added to increase system capacity; for example, a machine can be added to stage 2 of Configurations d as shown in Fig. 3d.

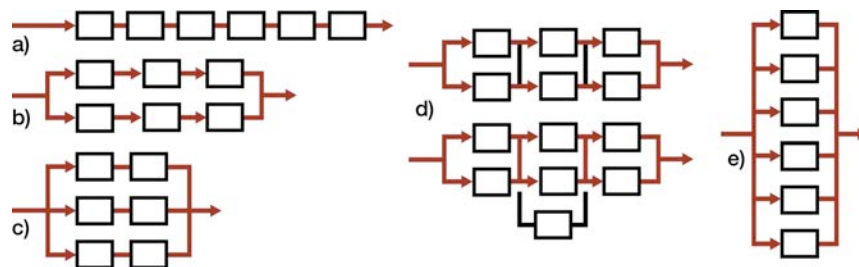


Fig 3. Five scalable configurations

In this example, the configuration depicted in Fig. 3c of two stages with three machines per stage, might be a compromise between reasonable scalability and investment cost. In this case, if a product requires machining on both the upper and side surfaces, the three machines in the first stage might be 3-axis vertical milling machines, and the three

machines in the second stage might be 3-axis horizontal milling machines. Conversely, in a parallel system, all six machines in Fig. 3e must be 5-axis milling machines – making the system much more expensive. In the system in Fig. 3c, capacity scalability must be performed in steps of 33.3% by adding one vertical machine and one horizontal machine, rather than in steps of 16.6% as with the parallel configuration. Adding a step of 16.6% in Fig. 3e in practice means adding one 5-axis machine with a large tool magazine that does the whole part processing.

To conclude, in general, the smallest scalability adjustment steps can be accomplished when the original system is purely parallel (e.g., Fig. 3e). However, the initial cost of a parallel system is the highest of all system configurations. In parallel configurations, each machine must perform all the manufacturing tasks needed to complete the part. Therefore, each machine must have the entire set of tools needed to produce the whole part and should also be able to perform more functions, for which more axes of motion are needed. As a result, the capital cost per additional volume increment added to a parallel configuration is the highest of all configurations.

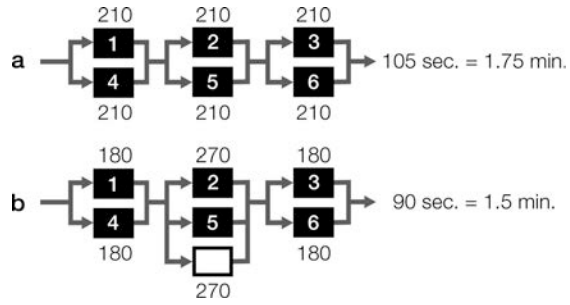
The following example clarifies the option of adding a small incremental capacity.

Example: On a system composed of six machines, as shown in Fig. 4, we have to process a part that requires 21 machining tasks of 30 second each, totaling 630 seconds, or 10.5 minutes, needed to machine each part. The required demand is 274 parts per 8-hour shift, namely 480 minutes. Therefore, the required cycle time is $480/274= 1.75$ minutes/part.

- a. Design a scalable system configuration.
- b. After one year, the demand has grown, and 320 parts per shift are needed, reducing the cycle time per part to 1.5 minutes/part. How many machines should be added, and what is the new configuration?

The cost-effective scalable system configuration is depicted in Fig. 3d, which is shown in detail in Fig. 4. Here, each machine does seven tasks, totaling 210 seconds per machine. When the demand grows to 320 parts/day, seven machines are needed. Only Configuration d yields the cost-effective solution by adding the new machine to Stage 2. One task is shifted from Stage 1 to Stage 2 so each machine in Stage 1 operates for 180 seconds on the part, and another task is shifted from Stage 3 to Stage 2; each machine in Stage 2 will then operate for 270 seconds on each part, as shown in Fig. 4b.

Fig. 4. When demand grows, the initial system, *a*, is cost-effectively scaled-up to Configuration *b* to meet the new demand



The initial capital investment in the system configuration in Fig. 4 is a bit higher than one in a serial line, because the material handling system is more complex. However, the extra capital investment is similar to buying an insurance premium for a future event which is likely to occur. If the demand does rise, the system can easily be scaled up and the new demand can be supplied in a short time, at a minimum additional investment. If the demand is unchanged during the lifetime of the system, a small capital investment on the more sophisticated material handling system was lost.

In this example, twenty-one equal tasks were needed to complete the part and the system with three stages and two machines per stage was perfectly scalable. In general, we will obtain similar scalability results in a symmetric configuration which has m stages and n machines per stage if the number of equal tasks needed to complete the part is:

$$(n.m + 1)m.$$

System design Fig. 4 the system designers must leave an empty space reserved for possible addition of a seventh machine and an extended material transport system to the spot.

3. Formulation of Scalability Planning Problem

We propose below a method to determine the most cost effective system reconfiguration to meet a new market demand. To perform system scalability planning, many factors need to be taken into consideration. These include a detailed process plan, setup plan, the machine capability, and the number of spots reserved for adding machines at each stage of the original system configuration. When reconfiguring an existing manufacturing system, simultaneous reconfiguration planning and system rebalancing attempts are needed to maximize the capacity of systems. In this section, an optimization model is proposed for the scalability planning. The solution to the model will be discussed in section 4.

3.1 Assumptions

The following assumptions are made based on the current industry practice in the powertrain industry.

- A multi-stage system with configuration as shown in Fig. 1 is considered. Parts are moved from one stage to another through conveyors and delivered to different machines within a stage using gantries.

- The number of stages will remain unchanged during the reconfiguration process. This way, the system setups will not be changed to avoid adjustment of the process plan, thereby minimizing impacts on the product quality.
- All the machines within the same stage perform exactly the same tasks.
- There are reserved spaces to add new machine in each stage and the gantries can be extended to deliver parts to the newly added machine(s).

3.2 Inputs

Scalability planning requires the following four types of inputs:

- Configuration information
 Number of stages L ;
 Number of machines in each stage N_i , where $i = 1, 2, \dots, L$;
 Maximum number of machines allowed in each stage M_i , $i = 1, 2, \dots, L$, which is restrained by the capability of material handling system.

- Stage Characteristics
 Each manufacturing stage usually has limited capabilities which are defined by a group of key characteristics of the stage. These include machine tool capability such as functionality, power, accuracy and machining ranges, and fixture capability such as face accessibility, which defines the faces that are accessible by the cutting tool. When a set of tasks are assigned to a stage, the necessary capabilities must fall into the key characteristics of the stage, otherwise the task allocation is invalid.

Assuming the number of key characteristics of each stage is K , a capability matrix S stores all possible key characteristics of each stage.

$$S[i, j] = d : d \text{ is } j_{\text{th}} \text{ key characteristic of stage } i, \text{ where } 1 \leq i \leq L, 1 \leq j \leq K$$

- Manufacturing tasks
 - *Task Precedence Tree*: manufacturing tasks must be performed at a certain order. This tree defines sequential constraints between tasks. Each task in the precedence tree can only be performed after all its parent tasks have been completed. A two-dimension binary matrix $Pre[1 \dots N, 1 \dots N]$, where N is the number of tasks to be processed, is used to represent the precedence tree.

$$Pre[i, j] = \begin{cases} 1, & \text{if task } i \text{ must be performed before task } j \\ 0, & \text{otherwise} \end{cases}$$

- *Task Key Characteristics*: These include task type, access direction, dimension, accuracy and power needed to perform the task. For a task to be assigned to a stage, its key characteristics must fall in the key characteristics of the stage. Assuming the number of key characteristic of each task is R , a task key characteristic matrix K is used to store the key characteristic of each task.

$$K[i, j] = f : f \text{ is the } j_{\text{th}} \text{ key characteristic of task } i. \text{ Where } 1 \leq i \leq N \text{ and } 1 \leq j \leq R.$$

- Machine reliability information
Machine reliability can be expressed by two parameters: MTBF (Mean Time Between Failure) and MTTR (Mean Time to Repair).
- Demand
The system will be reconfigured so its new capacity will fulfill the new demand D_{new} .

3.3 Decision variables

- Machine allocation array
 $M[i]$ = Number of machine being added to stage i , $1 \leq i \leq L$
- Task Allocation Array
 $T[i] = s$, s is an index of stage to which the task i is assigned, $1 \leq s \leq L$.

3.4 Mathematical Model

The following mathematical model is formulated to find the minimum number of machines to be added to the original system and the task allocation scheme for the reconfigured system to meet the new market demand.

$$\text{Minimize} \left(\sum_{i=1}^L M[i] \right) \quad (1)$$

Subject to

(1) Precedence constraints:

$$Pre[i, j] \cdot T[i] \leq T[j], \quad \forall i = 1, 2, \dots, N, \forall j = 1, 2, \dots, N \quad (2)$$

For two tasks i and j , if task i must be performed before j ($Pre[i, j] = 1$), it will be assigned to the same or a preceding stage as j .

(2) Key characteristic constraints

$$\prod_{k=1}^{NKC} (S[T[i], k] - K[i, j]) = 0, \quad \forall i = 1, 2, \dots, N, \forall j = 1, 2, \dots, R \quad (3)$$

Eq. (3) means that if task i is assigned to stage s , its key characteristics must fall in the key characteristics of stage s .

(3) Space constraints

The number of machines added to each stage must not exceed the limit.

$$M[i] \leq M \max[i], \quad \forall i = 1, 2, \dots, L \quad (4)$$

4 An Optimal Solution Approach Using Genetic Algorithm

A Genetic Algorithm (GA) is a structured heuristic that searches for good solutions using a mechanism that mimics “survival of the fittest.” Mixed integer problems with complex functions and combinatorial explosion on feasible space, like a scalability planning problem, can be solved efficiently by GA [24].

In a GA, populations of solutions are randomly initiated first. These are represented in a string format and will be evaluated to provide some measure of their fitness. This initial

population is then propagated into future generations by applying genetic operators to its members. After a number of generations, the algorithm converges to the near-optimal solution to the problem [25].

4.1 String Presentation

The first step of applying a GA is to represent a possible solution to the problem by the format of a string, namely encoding. A scalability scheme is encoded by an integer string which consists of two portions, corresponding to both an allocation scheme of manufacturing tasks, and system configuration.

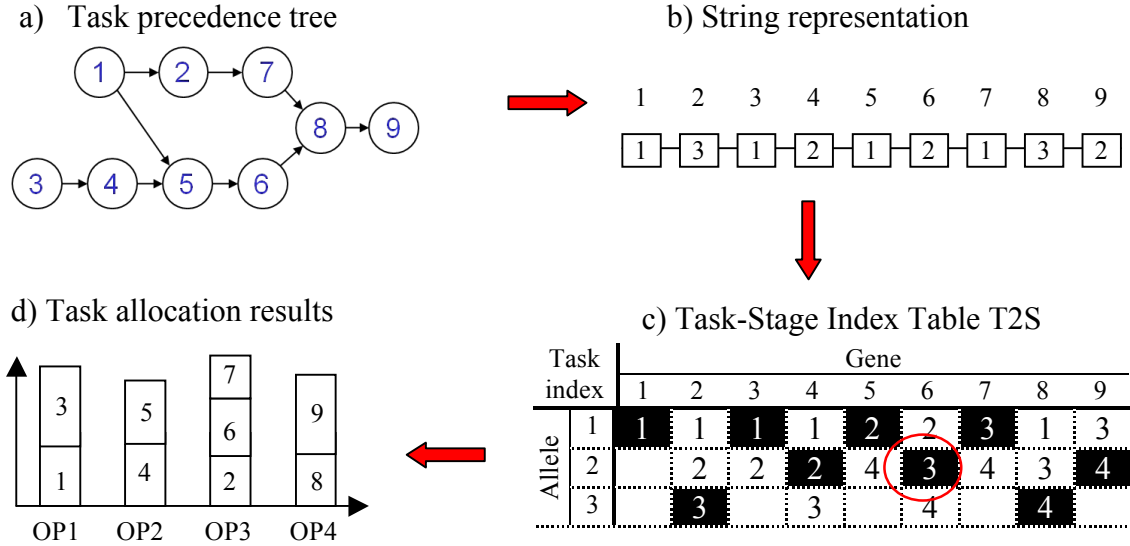


Fig. 5 String representation of task allocation

Fig. 5 gives examples of string representation of manufacturing tasks designed for task allocation. The length of each string is equal to the number of manufacturing tasks that need to be assigned to the manufacturing system. Each digit of the string contains two kinds of information: the position, *gene*, and the value, *allele*. The position indicates the sequential number of a task, and the value indicates the stage index which decides what stage the task will be assigned to. In this example, nine tasks are needed to complete a part in a system with four stages. They must be performed in the sequence depicted in Fig. 5a. An integer string with nine digits is used to present the nine tasks. For example, the first digit represents task 1 and the *n*th digit represents task *n*.

To ensure all the constraints are satisfied when assigning tasks to stages, a task-to-stage index table, or T2S as shown in Fig. 5c, is first built by comparing the key task characteristic to the key stage characteristic. The stage number can then be found in the T2S table according to the task number (*gene*) and the stage index number (*allele*). For example, the value of Task 6 in the string is 2, signifying it will be allocated to stage T2S [6, 2] which is OP 3 in the manufacturing system. Table T2S also determines the value range of each allele A_i which is the number of available stages that this task *i* can be assigned to. In the example of Fig. 5, the range of gene 1 is $A_1 = 1$; the range of gene 2 is $A_2 = 3$; the range of gene 9 is $A_9 = 2$ and so on.

During this decoding process, the following logic is used to ensure precedence constraints are not violated.

$$x(j) = \begin{cases} T2S(j, k), & \text{if } T2S(j, k) \geq x(i), \forall i, Pre(i, j) = 1 \\ T2S(j, k + \min(l)), & \text{until } T2S(j, k + l) \geq x(i), \forall i, Pre(i, j) = 1 \text{ and } \forall l = 1, 2, \dots, A_j \end{cases} \quad (5)$$

Where,

$x(j)$ is the stage number that task j is to be assigned to;

k is the allele value of task j ;

$T2S[j, k]$ is the stage number corresponding to allele k in task-to-stage index table;

$Pre(i, j) = 1$ indicates that task i is the parent task j ;

l is the smallest integer number that makes $T2S(j, k + l) \geq S(i)$, $\forall i$, for $Pre(i, j) = 1$

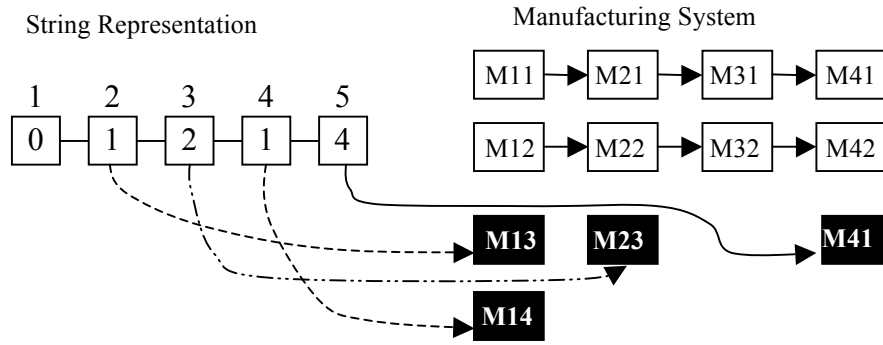


Fig. 6. String representation for system reconfiguration

An integer string representing system configuration, as shown in Fig. 6 is constructed in a way similar to Fig. 5. Assume a maximum of five machines are to be added to an original four-stage system to increase the system throughput for the new increased demand. The number of digits of the string is equal to the maximum number of machines to be added. The value of a digit represents the stage number a machine will be added to. A value of each digit varies from 0 to the number of stages. A “0” means an extra machine is not needed. When the number of machines added to one stage exceeds the maximum number of machines allowed by the material handlers, it is set to be equal to the maximum number of machines. For example in Fig. 6, the value of the first digit of the string is 0, means this machine is not needed; the value of the second digit is ‘1’, means this machine will be added to stage 1; the value of the fifth digit is “4”, means this machine will be added to stage 4.

The maximum number of machines is estimated as follows:

$$N_m = (D_{new} / P_{serial}) \cdot L - N_{M0} \quad (6)$$

Where,

L is the number of stages;

N_{M0} is the total number of machines in the existing system;

D_{new} is the new required system throughput;

P_{serial} is the system throughput of a serial line with L stages which can be calculated through line balancing.

4.2 Fitness (Objective) Function

This is a function which measures the relative worth of the solution when applied to a decoded string. In our case, the following fitness function is used:

$$f = \begin{cases} \sum_{i=1}^L M[i] + t_{s\max} / t_{\Sigma}, & \text{when } P \geq D_{new} \\ \sum_{i=1}^{LS} M[i] + t_{s\max} / t_{\Sigma} + Y, & \text{when } P < D_{new} \end{cases} \quad (7)$$

Where,

$\sum_{i=1}^L M[i]$ is total number of machines of the solution;

$t_{s\max}$ is the maximum stage cycle time of the solution, namely the cycle time of the bottleneck stage;

t_{Σ} is total task time;

P is the real throughput of the solution, which will be calculated using an ERC/RMS in-house developed software PAMS based on the analytical algorithm[26]; .

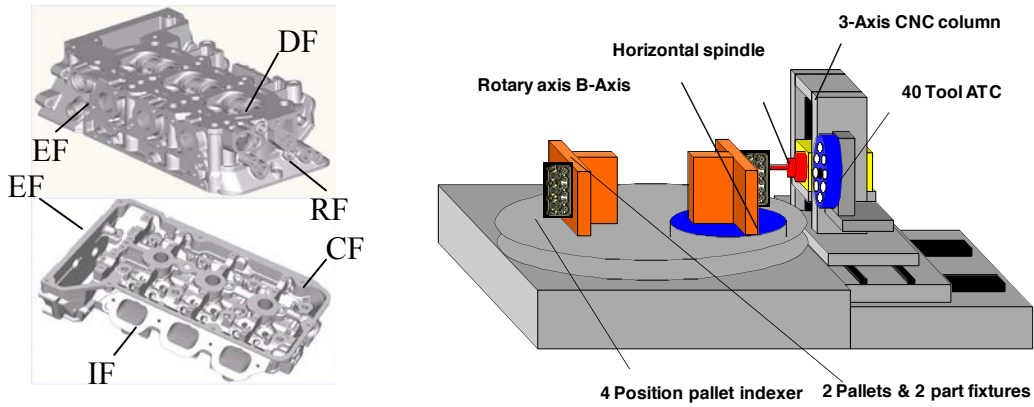
D_{new} is the new market demand;

Y is a penalty constant when the solution cannot fulfill the productivity requirement.

In equation (7), the first term is used to award the solution that yields the minimum number of machines. When two solutions give the same number of machines, the second term $t_{s\max} / t_{\Sigma}$ (always < 1) is used to award the solution which yields the most balanced results.

5. Case Study

This section describes experimental analyses employed to examine and validate the proposed approach for scalability planning. The case selected to validate our approach is the rough machining process of an automotive V6 cylinder head provided by an industrial partner to the NSF Engineering Center for the Reconfigurable Manufacturing Systems. There are 141 features on the cylinder head, which could be grouped into 43 machining tasks, including milling, drilling, boring, spot-facing, and tapping. Because of its complexity, this component was ideal for the study as it permits many process design solutions for different system configurations. The total time needed for the rough machining is 1019 seconds. The machines used for all stages are four-axis CNC machining centers which are capable of completing all the machining tasks. Fig.7a shows the part and Fig. 7b shows the machine configuration. MTBF and MTTR of the CNC machine are 193 minutes and 16.7minuts respectively. Fig. 8 shows a schematic diagram of the 43 machining tasks and the precedence tree used in this study.



(a) V6 Cylinder Head

(b) 4-axis CNC machining center

Figure 7 Part and machine configuration

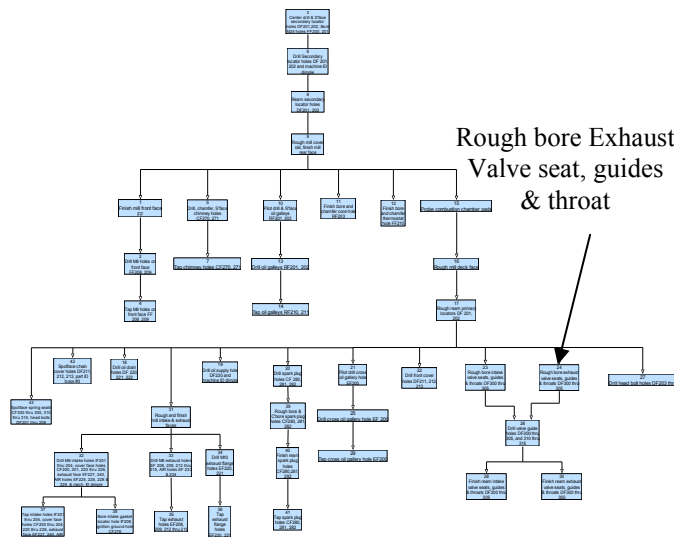


Figure 8 Task Precedence Tree

5.1 Baseline Configurations

Based on the requirements of the process plan of the part, the feasible setups are listed in Fig. 9. Different system configuration will need different combinations of these setups to produce the best processing results. Three system configurations, 3x4, 4x3 and 6x2, are used as the baselines to study scalability planning.

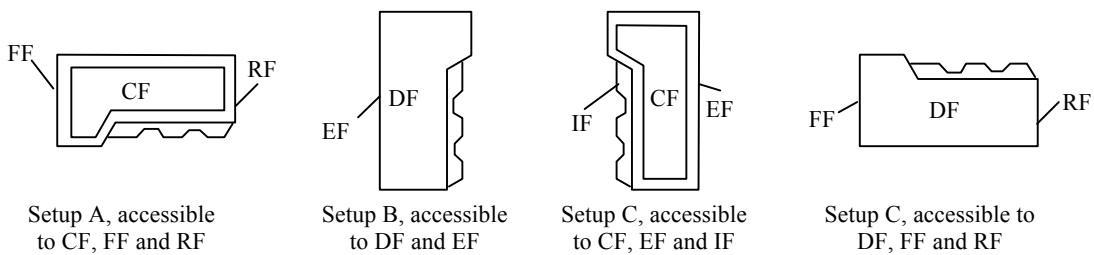


Figure 9 Feasible setups

Fig. 10 gives the three system configurations and their line balancing results. The letter under the OP number represents the setup number shown in Fig. 9.

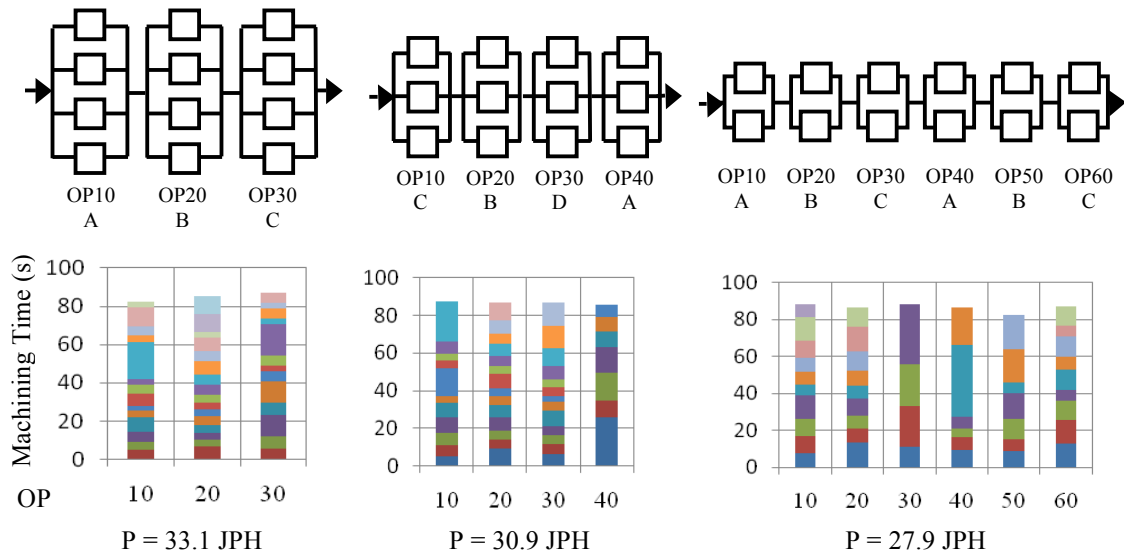
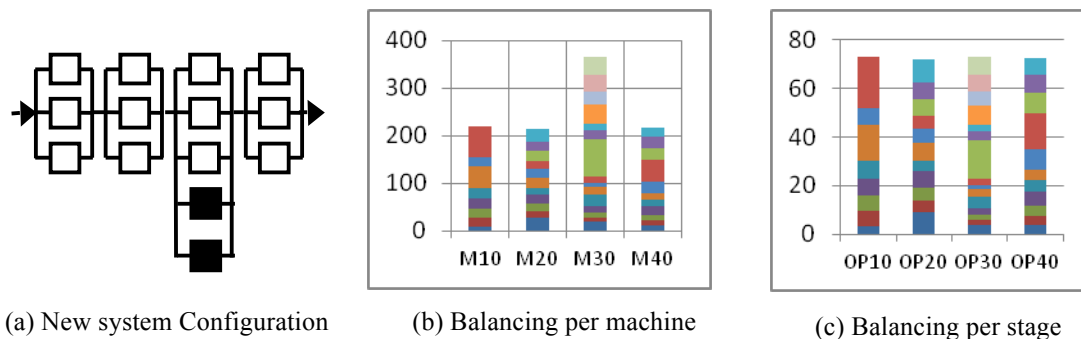


Figure 10 Three configurations and their productivity

5.2 Scalability Planning Results

Assume a 4x3 configuration (Fig. 10b) is currently being used to fulfill a production demand of 30JPH (jobs per hour). Also assume that a maximum of two machines can be added to each stage while the setup plan remains unchanged. When the new production demand changes to 35JPH, the proposed scalability planning algorithm found that 2 new machines need to be added to the system, as is shown in Fig. 11a. The rebalancing results per machine and per stage are shown in Fig. 11b and 11c, respectively. After adding two machines system capacity increased to 36.6JPH. Compared to duplicating a four machine serial line, the new configuration only needs two new machines to fulfill the new production demand.



(a) New system Configuration

(b) Balancing per machine

(c) Balancing per stage

System throughput increase to 36.6 JPH after adding two machines to stage 3

Figure 11 Scalability planning example of increasing productivity by 5JPH

Fig. 11 also shows that instead of being added to two different stages, the two machines are added to the same stage. This is because the selected setup plan and the machining tasks are not evenly distributed on each accessible face. Machines tend to be added to the

stages with a setup which allow access to more tasks. This way system through can be maximized.

For each configuration, reconfigurations for adding up to 5 machines to the existing systems are calculated. Fig. 12, 13 and 14 show the reconfigurations for a 3 stage system, 4 stage system and 6 stage system respectively. Again, Fig. 12 to 14 show that for a given case, machines are not evenly added to each stage. Some stages tend to require more machines than others to maintain the work load balance of the system. The number of machines and their locations to be added to the system can be optimized by the proposed method.

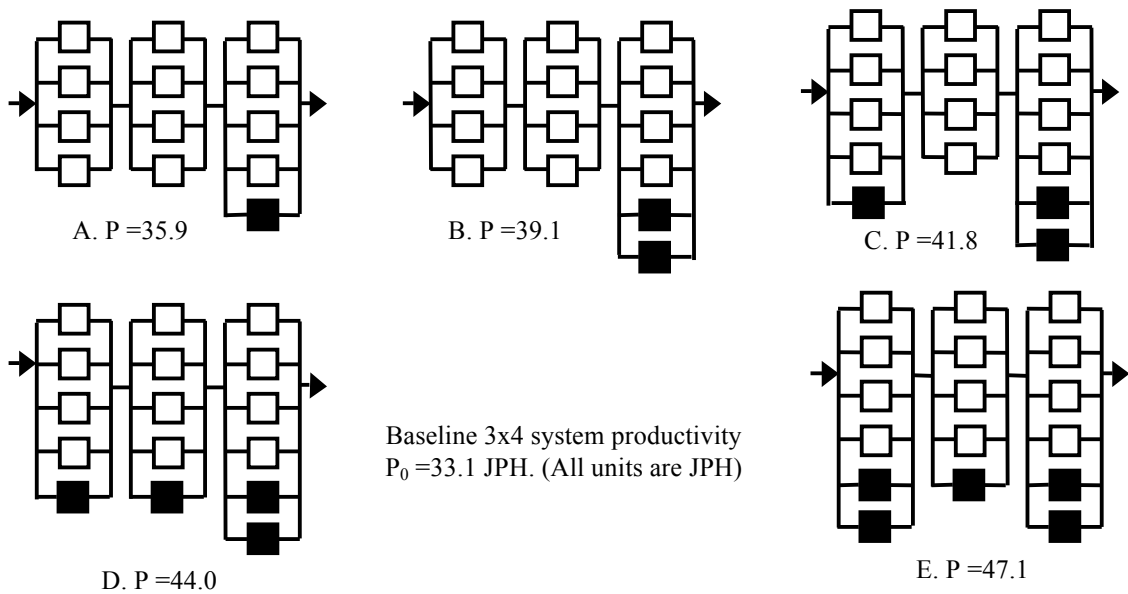


Figure 12 Reconfigurations for scalability planning for a 3x4 system

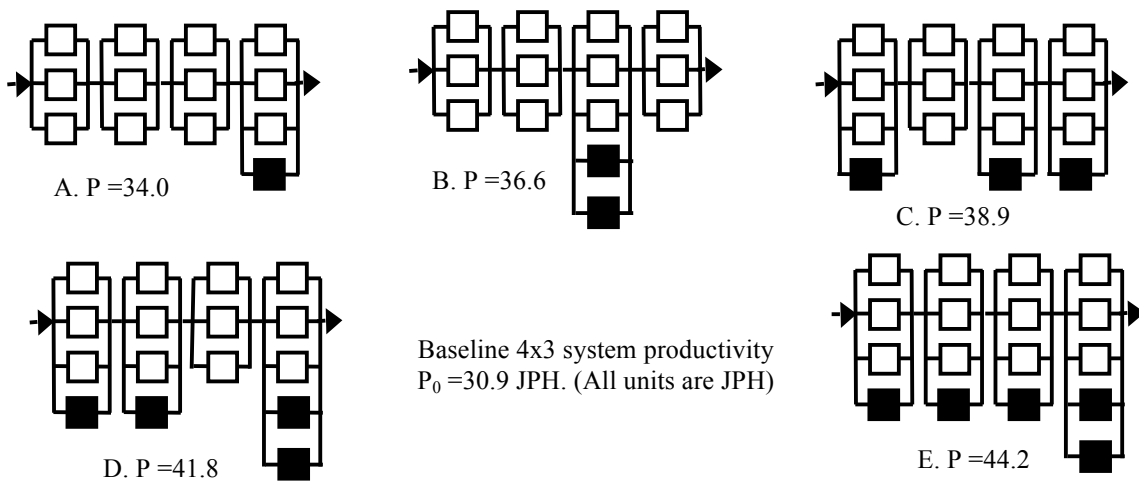


Figure 13 Reconfigurations for scalability planning for a 4x3 system

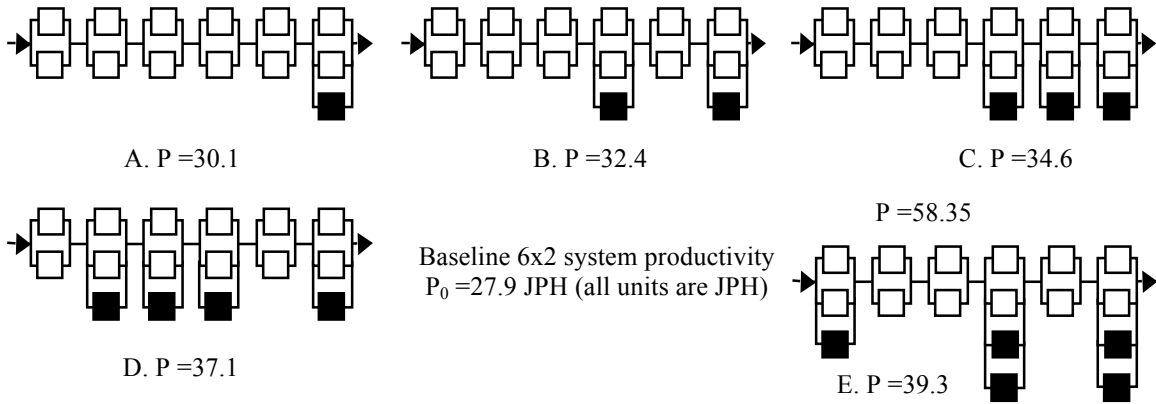


Figure 14 Reconfigurations for scalability planning for a 6x2 system

From the cost-effective point of view, we suggest scalability planning be performed concurrently with the design of a new manufacturing system. This way, optimal locations where future machines should be installed can be identified in advance. Thus, material handling system can be optimized for future scalability planning to reduce the life-time investment cost.

Table 1 summarizes the system productivity of each configuration and the new productivity when 1 to 5 machines are added to the existing systems. It can be seen that from Table 1 that system of 3x4 configuration gives both the largest system throughput and the largest throughput gain per machine. This is because both system reliability and system balance tend to decline with the increase of number of system stages. When the production demand increases for an existing system, Table 1, combined with Fig. 12 – 14, is very convenient for helping to decide how many new machines are needed and where they should be added to.

Table 1 System productivity of each configuration when new machines are added

<i>No. of Machines added</i>	System Throughput (JPH)						<i>Average throughput gain per machine</i>
	0	+1	+2	+3	+4	+5	
3x4	33.1	35.9	39.1	41.8	44.0	47.1	2.85
4x3	30.9	34	36.6	38.9	41.8	44.2	2.80
6x2	27.9	30.1	32.3	34.6	37.1	39.3	2.24

6. Conclusion and Discussion

This paper introduced the scalability concept and presented a systematic approach for scalability planning to add the exact capacity needed. This was done by simultaneously changing the system configuration and rebalancing the reconfigured system. An optimal solution approach, based on the Genetic Algorithm (GA), was developed for scalability planning with consideration of multiple constraints.

The proposed approach was examined and validated through a real industrial case. Experimental results showed the proposed approach can address the scalability planning

problem cost-effectively and efficiently. This paper suggests scalability planning should be performed concurrently with the design of a new manufacturing system. This way, the material handling system can be optimized for future scalability planning to reduce the investment cost.

For the purpose of simplicity, this paper only used the total number of machines as the optimization objective. However, in real production, many other cost factors need to be taken in consideration. These include labor, tooling, utility, floor space, operating cost, material handler, etc [22]. Since a reconfiguration process needs shutting down the production system which will also cause extra cost for production loss in addition to the abovementioned costs. This must be included into the optimization model for scalability planning in order to determine the optimal reconfiguration timing and how much capacity to add.

7 Acknowledgment

The writers gratefully acknowledge the financial support of the Engineering Research Center for Reconfigurable Manufacturing Systems (NSF Grant No. EEC95-92125) at the University of Michigan as well as the valuable input from the centers industrial sponsors.

References

1. Koren, Y. The global manufacturing revolution—product-process-business Integration and reconfigurable systems. John Wiley & Sons; 2010.
2. Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy AG, et al. Reconfigurable Manufacturing systems. CIRP Annals 1999; 48(2):6–12.
3. Koren, Y. and Shpitalni, M. Design of reconfigurable manufacturing systems. Journal of Manufacturing Systems 2010; 29 (4): 130–141.
4. Deif ,A. M., ElMaraghy, W. H. A Control Approach to Explore the Dynamics of Capacity Scalability in Reconfigurable Manufacturing Systems, Journal of Manufacturing Systems 2006; 25(1): 12-24.
5. Meng, X. Modeling of reconfigurable manufacturing systems based on colored timed object-oriented Petri nets. Journal of Manufacturing Systems 2010; 29(2-3): 81–90.
6. Lorenzer, T., Weikert, S. , Bossoni, S. , Wegenerb, K. Modeling and evaluation tool for supporting decisions on the design of reconfigurable machine tools, Journal of Manufacturing Systems 2007; 26 (3-4): 167–177.
7. Maier-Sperdelozzi, V., Koren, Y., and Hu S. J. Convertibility Measures For Manufacturing Systems. CIRP Annal 2003; 52(1): 367 – 371.
8. Michalek J., Ceryan O., Papalambros P. and Koren Y. Manufacturing Investment And Allocation In Product Line Design Decision-Making. ASME journal of Mechanical Design 2006; 128(6): 1196-1204.
9. <http://erc.engin.umich.edu/>
10. Koren, Y., Hu SJ., Weber T. Impact of Manufacturing System Configuration on Performance. CIRP Annals 1998; 47: 689-698.
11. Koren Y., Ulsoy A. G. Reconfigurable Manufacturing System Having a Production Capacity, Method for Designing Same, and Method for Changing its Production Capacity. US patent # 6,349,237. Issue date 2/19/2002. .

12. Son, SY., Olsen, T. L. Yip-Hoi, D. An approach to scalability and line balancing for reconfigurable manufacturing systems. *Integrated Manufacturing Systems*; 2001 12(7): 500 – 511.
13. Spicer, P., Koren, Y., Shpitalni M. Design Principles for Machining System Configurations. *CIRP Annals* 2002; 51(1):276–80.
14. Spicer, P., Yip-Hoi, D., Koren Y. Scalable Reconfigurable Equipment Design Principles. *International Journal of Production Research* 2005; 43(22): 4839 – 4852.
15. Zhang, D., Wang, L., Gao, Z. An integrated approach for remote manipulation of a high-performance reconfigurable parallel kinematic machine. *Journal of Manufacturing Systems* 2010; 29(4): 164–172.
16. Youssefa, A., and ElMaraghy, H. A. Performance analysis of manufacturing systems composed of modular machines using the universal generating function. *Journal of Manufacturing Systems* 2008; 27(2): 55-69.
17. Ariafera, S, and Ismail, N. An improved algorithm for layout design in cellular manufacturing system. *Journal of Manufacturing Systems* 2009; 28 4(): 132–139.
18. Um, I. Cheon, H., Lee, H. The simulation design and analysis of a Flexible Manufacturing System with Automated Guided Vehicle System. *Journal of Manufacturing Systems* 2009; 28 (4) pp. 115–122.
19. Nazarian, E. Ko, J. , Wang, H. Design of multi-product manufacturing lines with the consideration of product change dependent inter-task times, reduced changeover and machine flexibility. *Journal of Manufacturing Systems* 2010; 29 (1): 35–46.
20. Wang, H. , Zhu, X., Wang, H., Hu, S. J., Lin, Z., Chen, L.: Multi-objective optimization of product variety and manufacturing complexity in mixed-model assembly systems. *Journal of Manufacturing Systems* 2011; 30 (1): 16–27
21. Brooke, L., GM launches first CNC-based engine plant. *Automot. Ind.*, 1998, 128, 23
22. Freiheit, T., Wang, W. and Spicer, P. A case study in productivity-cost trade-off of production line parallelism. *International Journal of Production Research* 2007; 45(14): 3263-3288
23. DeGarmo, E., Black, J.T. and Kohser, R. *Materials and processes in manufacturing*, 7th ed., Macmillan, London, 1998
24. Hadj-Alouane, A., Bean, J. and Murty K. A hybrid genetic/optimization algorithm for a task allocation problem, *Journal of Scheduling* 1999; 2(4):189-201.
25. Gen M., Cheng R. *Genetic algorithms and engineering optimization*, Wiley, New York, 2000
26. Yang S., Wu C., Hu S.J. Modeling and analysis of multi-stage transfer lines with unreliable machines and finite buffers, *Annals of Oper. Res.* 2000; .93(1-4): 405-421.