Dynamism in a Semiconductor Industrial Machine Allocation Problem using a Hybrid of the Bio-inspired and Musical-Harmony Approach

Umi Kalsom Yusof 1 and Mohd Nor Akmal Khalid 1
1 School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia
E-mail: umiyusof@cs.usm.my; mnak104041@student.usm.my

Abstract. Semiconductor industries need to constantly adjust to the rapid pace of change in the market. Most manufactured products usually have a very short life cycle. These scenarios imply the need to improve the efficiency of capacity planning, an important aspect of the machine allocation plan known for its complexity. Various studies have been performed to balance productivity and flexibility in the flexible manufacturing system (FMS). Many approaches have been developed by the researchers to determine the suitable balance between exploration (global improvement) and exploitation (local improvement). However, not much work has been focused on the domain of machine allocation problem that considers the effects of machine breakdowns. This paper develops a model to minimize the effect of machine breakdowns, thus increasing the productivity. The objectives are to minimize system unbalance and makespan as well as increase throughput while satisfying the technological constraints such as machine time availability. To examine the effectiveness of the proposed model, results for throughput, system unbalance and makespan on real industrial datasets were performed with applications of intelligence techniques, that is, a hybrid of genetic algorithm and harmony search. The result aims to obtain a feasible solution to the domain problem.

1. Introduction
The semiconductor industry, because of its technologies and complex manufacturing procedure, has been widely regarded as having one of the most complex manufacturing processes [1]. Its nature of business, which is highly dependent on fast-growing and highly competitive technologies, requires the industry to invest on advanced state-of-the-art machines and facilities. At the same time, it also faces a continuous demand to reduce prices for its customers. Consequently, the industry encounters a great challenge in balancing the capital investment and the risk of losing customers. To maintain or regain competitive advantages, many semiconductor manufacturers have put more emphasis on improving capacity planning as well as scheduling and production control aside from having a good method of forecasting product demands.

Capacity planning is a process of adjusting one’s capability in response to the changing demands by maximizing machine utilization and allocating available resources to the right product mixtures while mitigating the technological constraints in meeting the customer demand. The ability to utilize the available resources to their utmost potential can be translated to a better managed and profitable company. Generally, the proposed approaches to solving the FMS machine allocation problem can be classified as (a) mathematical approaches, (b) heuristic
approaches, and (c) artificial intelligence-based (AI) or metaheuristic approaches [2]. Although mathematical approaches are robust in their applicability, they tend to become impractical when the problem size increases. Heuristic and AI-based approaches are more promising in terms of optimality and efficiency, even in more complicated problems, although sometimes the results are mainly dependent on the rules and constraints of individual problems.

These limitations motivate researchers to enhance the method and to look for innovative searching techniques to improve heuristic-based approaches further. One of the popular approaches is genetic algorithm (GA), which is known for its capability in intelligent probabilistic searching. [3] suggest the use of a GA-based scheduling tool to handle multiple resource constraints and a multiple-level product structure. [4] proposed a constraint-based GA in handling complex constraints in the FMS loading problem.

Another potential algorithm that can handle the machine allocation problem effectively is harmony search (HS). This heuristic algorithm is derived from an artificial phenomenon found in musical performance that always seeks for better harmony. HS has been applied to several computational optimization problems such as music composition [5] and timetabling problem [6]. According to [7], the main features of HS that make it different from other methods are as follows: HS makes a new vector after considering all existing vectors instead of considering only two (parents) as in GA, and HS does not require the setting of initial values of decision variables. These features increase the flexibility to find better solutions. In addition, [8] performed a study which concludes that HS poses a strong explorative power, which is a very important characteristic of evolutionary algorithms (EA).

According to [9], most studies associated with machine allocation problem are based on the assumption of a deterministic environment. However, in the actual manufacturing world, FMS operates in a dynamic environment where interruptions such as machine breakdown or introduction of new jobs are common and occur almost every day. Machine breakdown is one of the major undesirable inputs in the FMS environment. It causes not only additional maintenance cost but also increases the manufacturing lead time. According to an estimation by [10], maintenance cost alone can form 15% to 40% of the total production cost. In any manufacturing plant, the operating cost largely comes from the tools expenditure, which may take up to 30% of the total revenue [11]. Hence, it is important to minimize the negative effects of machine breakdowns because they may erode the profitability and productivity of FMS.

Consequently, it becomes critical to analyze and to minimize the adverse effect of failures on the objective measures of the machine allocation problem so that production goals can be achieved. Only a few studies have been conducted on the machine allocation problem that deals with machine interruption and control policies, mimicking the real environment of FMS. In this study, the real shop floor scenario is emulated, that is, the handling the machine allocation problem with the occurrence of a machine breakdown that aims to minimize the effect of the machine breakdowns on the overall performance of FMS. The model that encompasses an online machine monitoring scheme is proposed to handle the expected breakdowns due to wear-outs and manual machine monitoring, dealing with random failures that may involve the reloading of part types. In the online monitoring scheme, machines are continuously monitored to measure the failure potential and determine the action beforehand to avoid a potential breakdown. Conversely, manual monitoring ensures that the action is taken as soon as possible to minimize the effect due to a sudden breakdown.

In studying the well-known approaches to dealing with the conventional machine allocation problem, some of the researchers preferred using of a probabilistic search technique to gain optimum result. Meta-heuristic algorithms such as GA and HS are among the effective random search techniques, that can produce a near optimum result from a large solution space of the machine allocation problem in an acceptable computational time. Thus, these computational techniques have been employed to obtain quality solutions under machine breakdown conditions.
With the added advantage of GA and HS combined together in a hybrid; good results can be obtained.

Our work is an extension of the results obtained from the previous works on constraint-chromosome genetic algorithm (CCGA) [12], HS [13], and a hybrid of CCGA and HS (called H-CCGaHs) [14] performed on the benchmark problems available in [15]. For a detailed explanation on how the H-CCGaHs and the related operators work, refer to the previous related works. Based on the result, the proposed algorithm H-CCGaHs performs better than most of the heuristics available in the literature in the minimization of system unbalance as well as combined objective functions (COF).

The proposed H-CCGaHs can achieve the best solution for 7 out of 10 datasets tested as opposed to only 6 datasets for CCGA and HS. The average COF (1.8160) in the last row of Table IV in [14] shows that H-CCGaHs is superior in performance to the other heuristics considered in this evaluation. This paper focuses on the application of the proposed H-CCGaHs that incorporates machine breakdowns with the semiconductor industrial machine allocation problem using actual industrial datasets. As the nature of the datasets used in [14] is almost similar to that of the actual industrial datasets (except that the latter datasets are much bigger), this algorithm should be able to produce the same excellent result if experimented on actual industrial datasets.

The remainder of the paper is organized as follows. Section 2 describes the dynamic machine allocation, and Section 3 formulates the model formulation. Section 4 discusses the proposed solution to solve the problem, and Section 5 presents the results of the proposed solution. Section 6 concludes the paper.

2. Industrial Dynamic Machine Allocation Problem

One of the concerns of the dynamic machine allocation problem is the determination of the optimal sequence of part types considered for production. The solution should consider the unavailability of the machines that were previously allocated. Each part type is characterized by a set of operations with a pre-specified processing time. The objective is to allocate the operations to a set of machines such that some quantifiable measurements can be set up and optimized [4]. These measurements include the minimization of makespan and system unbalance and the maximization of throughput, considering the various technological constraints related to machining time.

Part type selection and machine allocation arrangement constitute two major components of the strategic planning of FMS. Part type selection deals with selection of a set of part types to be manufactured during the upcoming planning horizon, and the loading problem concerns the allocation of operations and the required machines and tools for the selected part types. Most of the earlier researchers managed these problems separately because of their complexities. Therefore, the solution of the part type selection problem may lead to an infeasible result for the loading problem.

There are two types of operations according to each part type. Essential operations of a part type indicate that these operations can be performed only on a particular machine using a certain number of tool slots, whereas optional operations imply that they can be carried out on a number of machines with the same or varying processing time and tool slots. As essential operations require performance at specific machine/s, flexibility lies in the selection of machines that are set for optional operations in which the machine allocation can be improved to give better results.

In this research, the dynamic machine allocation problem was solved in the context of random-type FMS. With the various part types being loaded and having a wide range of processing requirement variations, the decision on the machine allocation should be dynamic as well. The assembly line typically has machines that normally consist of multiple-functional machines,
and fixed processing time is associated with it depending on the part types to be assembled. Depending on the product build specifications, certain operations of the part types can be essential, optional, or a combination of both. The essential operations are performed on a specified machine, whereas optional operations can be performed on any of the alternative machines. To denote the different handling of part types during machine breakdown, the part types in this study are categorized as follows [9]:

- Regain part type: In this category, the processing of interrupted operation (essential or optional) is resumed using the same machine immediately after it has been repaired. The processed portion of the job before the breakdown is considered valid.
- Repeat part type: In this case, as it involves the transfer of the job to a different machine, the job before the breakdown is discarded, and the whole operation (essential or optional) has to start from the beginning.

Machine breakdown is categorized into two types, namely, predetermined machine breakdown (PMB) and stochastic machine breakdown (SMB).

3. Model Formulation

3.1. Subscripts and parameters notations

The subscripts and parameters notations used to demonstrate the objective functions are as follows:

Subscripts
- \( N \): Total number of parts
- \( M \): Total number of machines
- \( H \): Planning horizon
- \( i \): Part type, \( i = 1, \ldots, N \)
- \( m \): Machine, \( m = 1, \ldots, M \)
- \( j \): Operation type, \( j = 1, \ldots, J \)

Parameters
- \( U_{T_m} \): Under-utilized time on machine \( m \)
- \( O_{T_m} \): Over-utilized time on machine \( m \)
- \( \beta_i \): Batch size of part type \( i \)
- \( t_{a_{im}} \): Time available on machine type \( m \) after the allocation of operation \( j \) of part type \( i \)
- \( t_{imj} \): Time required by machine type \( m \) for operation \( j \) of part type \( i \)
- \( T_{r_{im}} \): Number of machines for each machine type remaining on machine \( m \) after the allocation of operation \( j \) of part type \( i \)
- \( T_{m} \): Number of machines for each machine type available on machine \( m \) for operation \( j \) of part type \( i \)
- \( T_{imj} \): Number of machines for each machine type required by machine type \( m \) for operation \( j \) of part type \( i \)
- \( Y_{ij} \): Set of machine types in which operation \( j \) of part type \( i \) can be performed after the allocation of operation \( j \) of part type \( i \)
- \( R_{p_{im}} \): Repair duration for predetermined breakdown on machine \( m \)
- \( R_{s_{im}} \): Repair duration for stochastic breakdown on machine \( m \)
- \( T_{s_{imj}} \): Start time of operation \( j \) of part type \( i \) on machine \( m \)
- \( T_{e_{imj}} \): End time of operation \( j \) of part type \( i \) on machine \( m \)
- \( T_{sb_{imj}} \): Time at which stochastic breakdown occur processing of operation \( j \) of part type \( i \) on machine \( m \)
- \( T_{s} \): Slack time \( (T_{sb_{imj}} - T_{s_{imj}}) \)
3.2. Objective functions

The following equation calculates the cumulative processing time of a machine under breakdown circumstances:

\[ CP_m = \sum_{i=1}^{N} \sum_{k=1}^{J} \alpha_i \left( (\alpha_i \ast t_{imj}) + (Bp_{imj} \ast R_m^i) + Bs_{imj}(\theta_i + R_m^i + \delta_i \ast (Ts + Bs_{imj} - t_{imj})) \right) \]  

\( CP_m \) (cumulative processing time) is the working time for machine \( m \). Equation 1 clearly shows the negative effect of a breakdown, as it increases the working time. In the event of PMB or regain SMB, the makespan is increased because of the amount of repair time. In the event of repeat SMB, the makespan is increased because of repair time and slack time \( Ts \) (as the affected part types are transferred to the nearest available machine for reprocessing).

(i) System unbalance (SU) is the summation of the remaining time or idle time in all available machines. By minimizing SU, the utilization level of the machines will increased. This objective function is represented by notation \( F_1 \).

\[ \arg \min_{F_1} F_1 = \left( \frac{SU_{seq}}{SU_{max}} \right) \]  

which means

\[ \arg \max_{F_1} F_1 = \left( \frac{SU_{max} - SU_{seq}}{SU_{max}} \right) \]  

where

\[ SU_{max} = \sum_{m=1}^{M} mtm_m \]  

\[ SU_{seq} = \sum_{m=1}^{M} rtm_m \]  

\[ rtm_m = mtm_m - CP_m \]  

and \( mtm_m \) designates the total available time of machine \( m \), and \( rtm_m \) indicates the remaining time on machine \( m \). Equation 6 proves that, in the event of PMB and regain SMB, machine available time is reduced by both processing time and repair duration. In the event of repeat SMB, aside from the processing time and repair duration, available time in the machine also decreases because of slack time, as the disrupted part type is transferred to the nearest available machine within the same group for re-machining.

(ii) Throughput is defined as the summation of the batch sizes of the selected part types. More importantly, the maximization of throughput increases the productivity of manufacturing systems.

\[ \arg \max_{F_2} F_2 = \left( \frac{TH_{seq}}{TH_{max}} \right) \]  

where
The notation $T H_{\text{max}}$ and $T H_{\text{seq}}$ represent the throughput of a particular sequence of part types and the maximum throughput (summation of batch size of all part types), respectively.

(iii) The makespan or the cumulative processing time of machines, $F_3$, is defined as follows:

$$F_3 = \arg\min_{F_3} \left( \frac{\max [CP_m]}{\sum_{i=1}^{N} \sum_{j=1}^{J} t_{ij}} \right),$$

which means

$$\arg\max_{F_3} \left( \frac{\sum_{i=1}^{N} \sum_{j=1}^{J} t_{ij} - \max [CP_m]}{\sum_{i=1}^{N} \sum_{j=1}^{J} t_{ij}} \right),$$

where $t_{ij}$ denotes the processing time of operation $j$ of part type $i$ (either selected or rejected).

(iv) The fourth objective function is a combination of all the above objective functions:

$$\arg\max_{F} F = \left( \frac{W_1(F_1) + W_2(F_2) + W_3(F_3)}{W_1 + W_2 + W_3} \right),$$

where

- $W_1$ Weight assigned to objective function $F_1$
- $W_2$ Weight assigned to objective function $F_2$
- $W_3$ Weight assigned to objective function $F_3$

For simplicity, 1 unit of weight is assigned to each of the objective although in real world, an individual objective may carry a different weight.

4. Proposed Solution

The part type sequence determines the processing arrangement of the long string of part types. The first part type in the sequence is allocated first, followed sequentially by the rest of the part types according to the sequence string. As the industrial dataset has a long sequence, maintaining the uniqueness of the part types takes a long processing time. Therefore, a method needs to be identified in which the processing time should not take too long or should be within the permissible processing time for it to be effective. One of the widely used methods is randomized algorithm in which the input (part type sequence array) is randomized through permutation array [16]. This random re-arrangement of the part types will not disturb or change their uniqueness, as each part type in the sequence is unique.

A sample of the part sequence for dataset CP is shown in Figure 1(a). Based on the sequence part type, the chromosome called the part-chromosome will be created. To ensure the feasibility of the solution, each gene in the part sequence will retrieve the operations assigned to it. As the operation sequence for each part type has to follow the sequence, indicating that the first
operation needs to be allocated first, followed by the second operation and so forth, maintaining the sequence is required.

Part types in a dataset can carry a large quantity that should be allocated to either one or more machine types. This requirement enables the design of the gene representing the machines to be allocated as a 4-bit string (maximum of 4 machine types) with “1” denoting the corresponding machine type being allocated, and “0” indicating otherwise (e.g. 1 0 0 1 means machines 1 and 4 being allocated). The number of genes that can be created is \(2^n - 1\).

For example, the operation with 2 machine types will have a maximum of three genes, and the operation with 4 machine types will have 15 possible genes.

Figures 1(b) and (c) two examples of the possible values of the part-operation chromosome based on the part sequence [1 5 3 2 A 4 N 6 O 9]. The first two digits of the values denote the part type and operation number, respectively, followed by the 4-bit string of machine types. As operation 1 of part type 4 is considered an essential operation and can only be performed on a particular machine, there is no change in gene that generates the part-operation-chromosome. Conversely, operation 4 of part type 2 is an optional operation that can be performed on a number of machines with the same or varying process times. In this example, operation 4 can be performed on all four machine types, in which the total processing times for these machine types could vary and lead to a different result.

5. Experimental Results and Discussion

The various combinations of the parameter values are used to conduct a sensitivity analysis of the sample dataset to identify the optimal control parameters for the proposed algorithms. [9] used a population size of 20 to 30 to handle 6 to 14 jobs. Using this as a baseline, for an average of 90 part types, the best population size is 300. In the same work, the crossover range between 0.7 and 0.8 produced better results for a large-sized problem. Therefore, 0.7 and 0.8 for the probability of crossover were tested. For the mutation, the range of 0.1 to 0.3 was taken. The number of generations was set to 75, as earlier experiments show that the increase in number of generations from 50 to 75 does improve the result. Table 1 shows the details of the control parameters for the algorithm.

Five datasets consist of 10 packages, denoted as CP, WP1, WP2, WP3, and WP4 containing 10, 105, 72, 105, and 140 number of parts, were used for the experiments. The CP dataset represents the production plan that intends to be loaded in four operations, and WP1, WP2, WP3 and WP4 represent the work-in-process lots that are already in production, requiring resources to be allocated for the processing of the corresponding operations. Owing to the nature of the dataset, the CP quantity becomes a lump sum of the quantity per package, and the machine processing hours are expected to be shared and loaded for the week (123.25 hours).
Datasets WP1 to WP4 are loaded in shifts (7.25 hours). Thus, the same quantity will have the possibility of being allocated with more than one type of machines, considering the fact that one machine type consists of many machines. These part types are scheduled to be processed using 141, 67, 13, 31, and 20 machines, respectively. To demonstrate the robustness of the proposed model of the dynamic machine allocation problem, the proposed H-CCGaHs was used on these five industrial datasets.

Table 2 shows the machine resource summary in which each operation has a number of machine types, and each machine type has a number of machines. Based on the number of machines and processing hours, weekly and shifting remaining time hours (RTM) are calculated, and shifting and weekly predetermined machine breakdowns are calculated based on the probability of PMB. Note that the machine unit processing per hour (UPH) is based on the operation number, machine type, and product type. The same machine type may vary its UPH if run on a different product type. The processing time allocated to perform the planned quantity is based on the total quantity over the UPH of the machine type.

**Table 1. Control parameters for the algorithm**

| Option | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------|---|---|---|---|---|---|---|---|---|----|----|----|
| Crossover Operator | Ordered Chromosome |
| Mutation Operator | Reciprocal Exchange |
| Selection Operator | RW | TN |
| Probability of Crossover | 0.7 | 0.7 | 0.7 | 0.8 | 0.8 | 0.7 | 0.7 | 0.7 | 0.8 | 0.8 | 0.8 |
| Probability of Mutation | 0.1 | 0.2 | 0.3 | 0.1 | 0.2 | 0.3 | 0.1 | 0.2 | 0.3 | 0.1 | 0.2 |
| HMCR | 0.85 | 0.9 | 0.95 | 0.85 | 0.9 | 0.95 | 0.85 | 0.9 | 0.95 | 0.85 | 0.9 |
| PAR | 0.01 | 0.1 | 0.2 | 0.2 | 0.01 | 0.1 | 0.01 | 0.1 | 0.2 | 0.2 | 0.01 | 0.1 |

Number of generations/iterations: 50; Population/Harmony Memory size: 300; RW: Roulette wheel; TN: Tournament

HMCR: Harmony memory consideration rate; PAR: Pitch adjustment rate

This random type FMS under consideration receives several part types with various processing requirements that are to be loaded on different machines. Consider that predetermined machine
breakdown occurs on every 1.5 shifts for each machine on average, which is approximately 660 minutes. Under the online monitoring scheme adopted, the maintenance department is assumed to take 30 minutes to repair the failed machines.

SMB is also assumed to occur on probability of 15% of all the machines available and that the maintenance personnel takes a maximum of an hour to diagnose the problem and repair the machine. In the case of the repair time being more than an hour, the machine will be shut down for further investigation, and the part types that are running on that machine will be transferred to a different machine. The probability of this scenario happening is estimated to be 6% of total machines. The transfer time and the set up on the new machine, including buy off is assumed to be 45 minutes on average.

The proposed approaches were implemented using C# compiler. Table 3 shows the results obtained by our proposed algorithm. TH, SU and MS for each dataset, together with number of part types allocated and the number of part types that are not allocated, are shown along with the fitness values. The nature of the datasets causes SU and MS to be higher in dataset CP, although the percentage of TH is relatively low. The reason is that the quantity to be processed for each part type in dataset CP is large and requires a huge block of machine available hours to accommodate such as a big number. As stated previously, the main difference between CP and WPs is that CP is a production plan schedule that consists of a bulk quantity to be processed in the range of the operations, and WPs are the work-in-process (WIP) datasets that are currently awaiting to be processed in the next operation. Based on these results, TH is reduced due to machine downtime as less machine available hours are allocated. Moreover, the numbers of part types assigned (i.e., number of PA) is also reduced.

| Dataset | Total Quantity | Total Avail. Time | Total Proc. Time | TH | SU | MS | COF | No. of part types | No. of PA | No. of PU | Convergence | Time to run |
|---------|----------------|------------------|-----------------|----|----|----|-----|------------------|----------|----------|-------------|------------|
| CP      | 10493472       | 17378.3          | 19518.4         | 5287.2 | 16686.9 | 1.5809 | 10 | 8 | 2 | 48 | 02:17.6 |
| WP1     | 746031         | 558.3            | 881.2           | 588300 | 1.0 | 710.7 | 1.9802 | 105 | 81 | 24 | 66 | 13:59.7 |
| WP2     | 364956         | 94.3             | 667.4           | 85181 | 0.0 | 667.4 | 1.2332 | 72 | 18 | 54 | 44 | 0:06:36 |
| WP3     | 681127         | 224.8            | 275.2           | 549521 | 120.4 | 223.1 | 1.4605 | 100 | 69 | 31 | 39 | 0:25:13 |
| WP4     | 994708         | 145.0            | 251.5           | 653640 | 0.2 | 212.2 | 1.8116 | 140 | 87 | 53 | 43 | 0:37:33 |

Tot. Avail. Time: Total available time; Tot. Proc. Time: Total processing time.
TH: Throughput; SU: System Unbalance; MS: Makespan; COF: Combined objective function value.
PA: Part assigned; PU: Part unassigned.

As reported from the previous study [14], H-C CGA Hs is a promising approach in to solving the machine allocation optimization problem because it produces the best result with reasonable time taken to complete the task and converge within an acceptable number of generations. The strength of H-C CGA Hs relies on the population swapping generated by both GA and HS and on the running against the other algorithm’s operators after swapping, thus creating more chances of discovering the unvisited solutions and avoid the creation of a repeated or similar solution. This finding indicates that the time taken to find solutions is not only dependent on the size of the population but also on the tightness of the constraints and the approach used in designing the chromosome representation as well as on the effectiveness of the fitness function. The reason is that a substantial processing time is required to filter non-feasible solutions.

6. Conclusion

Applying the same algorithm to the manufacturing environment can be very challenging. It has its own constraints. The size of the industrial datasets used is typically large. In fact, the sample
industrial datasets used in this experiments are only 5% to 10% of the actual manufacturing dataset size, and these are equivalent to more than 10 times of the benchmark datasets in [15]. Therefore, the chromosome representation tends to be very long. The part types in this experiment were denoted as a string of sequence, and no part type duplication was allowed. The sequence determines the processing arrangement of the part types during the initialization of the populations. As the industrial dataset has a long sequence, maintaining the uniqueness of the part types will take a large proportion of the processing time. To overcome this problem, RIP, one of the randomized array introduced by [16], was adopted. As presented in Table ??, the estimated running time is reduced by up to 510 times compared with using a conventional randomized method.

Inspired by the successful application of intelligent search techniques in the field of combinatorial complex problems, a hybrid of the H-CCGaHs was employed to determine the robustness of the proposed model of the dynamic machine allocation problem. A comparative and analytical study was performed. The excellent results obtained from the case study show that the proposed model can provide a good, effective, and practical solution for industrial application. Extending the research, the superiority of the algorithm was also tested on a larger-sized problem adopted in this research. Moreover, various analyses were performed that would enable the manufacturers to consider the approaches most suitable for their environment to meet their business goals. The results demonstrate the superiority of H-CCGaHs among the applied algorithms in terms of its ability to find the highest overall COFs within a reasonable running time. This work is expected to aid the management in better managing the machine utilization as well as enabling more part types to be scheduled using the current resources.

Acknowledgments
The authors wish to thank Universiti Sains Malaysia for the support it has extended in the completion of the present research through Short Term University Grant No: 304/PKOMP/6313026.

References
[1] Huang D, Yan J and Qiao F 2004 8th International Conference on Control, Automation, Robotics and Vision 3 2217–2222
[2] Nanvala H B and Awar G K 2011 International Journal of Engineering and Technology 3 64–73
[3] Pongcharoen P, Hicks C and Braiden P M 2004 European journal of Operational Research 152 215–225
[4] Kumar A, Prakash, Tiwari M K, Shankar R and Baveja A 2006 European Journal of Operational Research 175 1043–1069
[5] Geem Z 2006 Improved Harmony Search from Ensemble of Music Players (Lecture Notes in Computer Science vol 4251) (Springer Berlin / Heidelberg)
[6] Al-Betar M and Khader A 2012 Annals of Operations Research 194 3–31
[7] Geem Z W, Kim J H and Loganathan G 2001 SIMULATION 76 60–68
[8] Mukhopadhyay A, Roy A, Das S and Abraham A 2008 Third International Conference on Digital Information Management, 2008. ICDIM 2008. pp 775 –781
[9] Mandal S K, Pandey M K and Tiwari M K 2010 International Journal of Production Research 48 3535–3559
[10] Sheu C and Krajewski L 1994 International Journal of Production Research 32 1365–1382
[11] Calanaydirim M R R 2002 IIE Transactions 34 449–465
[12] Yusof U K, Budiarto R and Deris S 2012 International Journal of Innovative Computing, Information and Control 8 1591–1609
[13] Yusof U K, Budiarto R and Deris S 2011 Data Mining and Optimization (DMO), 2011 3rd Conference on pp 26–31 ISSN 2155-6938
[14] Yusof U K, Budiarto R and Deris S 2011 Proceedings of Sixth International Conference on Bio-Inspired Computing: Theories and Applications, Penang, Malaysia pp 89–96 ISBN 978-0-7695-4514-1
[15] Mukhopadhyay S K, Midha S and Krishna V M 1992 International Journal of Production Research 30 2213–2228
[16] Cormen T H, Elision C E, Rivest R L and Stein C 2008 Introduction to Algorithms (Prentice-Hall, Inc.)