

Agent-based Hierarchical Production Planning and Scheduling in Make-to-Order Manufacturing System

N. He^{a,1,*}, D.Z. Zhang^{a,2}, Q. Li^{b,3}

^aCollege of Engineering, Mathematics and Physical Sciences, University of Exeter,
Harrison Building, North Park Road, Exeter, EX4 4QF, UK

^bCollege of Mechanical Engineering, Chongqing University, Chongqing, 400044, China

Abstract

Nowadays, manufacturing organisations face increasing pressures from the frequent change in product type, continuous demand fluctuation and unexpected change in customer requirements. In order to survive in the turbulent environment, manufacturing organisations must become flexible and responsive to these dynamic changes in the business environment. This paper presents a hierarchical agent bidding mechanism that is particularly designed for Make-to-Order manufacturing system and attempts to enhance the operational flexibility of manufacturing system in dealing with dynamic changes in the business environment. The novelty of this mechanism is that it enables manufacturing resources to be self-organised cost-efficiently within structural constraints of manufacturing system for fulfilling customer orders. However, when orders cannot be fulfilled within the structural constraints of manufacturing systems, the mechanism can enable manufacturing resources to be regrouped flexibly across system boundaries but with minimum disturbances to existing system structure. Based on an example application to a manufacturing company, this paper demonstrates that the operational flexibility provided by this mechanism is able to help manufacturing system to respond demand fluctuation through balancing the capacity across the entire system. Meanwhile, this mechanism potentially enables manufacturing systems to deal with unexpected changes in product type. As long as the manufacturing system has the technicality required by a new product, this mechanism enables resources across the manufacturing system to be cost-efficiently and flexibly self-organised to fulfil the new product.

Keywords: Hierarchical Planning, Multi-agent System

*Corresponding author

Email addresses: N.He@exeter.ac.uk (N. He), D.Z.Zhang@exeter.ac.uk (D.Z. Zhang),
cuq.qiangli@gmail.com (Q. Li)

¹Tel.: +44(0)1392723363

²Tel.: +44(0)1392723641

³Tel.: +86 13668006620

1. Introduction

Nowadays, due to the shift of production modes from mass production to mass customisation, increase of customer knowledge, and rapid development of communication technology, manufacturing organisations face more and more pressures from the dynamic changes in the business environment e.g. frequent change in product type and unexpected change in demand pattern. Over the past decades, several approaches have been developed to assist manufacturing organisations in dealing with changes in the business environment. For example, manufacturing system concepts exist, which are designed to aid manufacturing organisations in responding to change in product type. These include Flexible Manufacturing System (FMS) (Stecke, 1983; Buzacott and Yao, 1986) and Reconfigurable Manufacturing System (RMS) (Koren et al., 1999). Both concepts stem from the introduction of advanced machines (i.e., flexible machines and reconfigurable machines) and aim to cope with change in product type through machine flexibility. However, they are not applicable to manufacturing systems with existing available resources due to a large amount of initial capital investment on resource replacement. Further, these concepts, particularly FMS, may create waste of investment because of inaccurate forecasting of future product types and customer requirements. Moreover, other changes in the business environment like change in demand pattern cannot be handled simultaneously by the use of these two concepts. In terms of changes in demand pattern, two approaches related to layout design have been developed, including the dynamic layout (Rosenblatt and HAU, 1987; Balakrishnan et al., 1992; Balakrishnan and Cheng, 1998; Kochhar and Heragu, 1999) and robust layout (Rosenblatt and Kropp, 1992; Benjaafar et al., 2002). These approaches aim to add built-in flexibility or robustness to system layout in response to forecasted changes in demand pattern. However, since either the layout flexibility or layout robustness is pre-determined based on demand pattern forecasting at the beginning of a long-term planning period, both approaches are unable to deal with unexpected changes incurred within the planning period. Also, as these two approaches only consider machine arrangement at the bottom manufacturing system level (e.g., manufacturing cells), they cannot be applied to manufacturing system with complex hierarchy and therefore do not necessarily enhance the overall system flexibility.

Consequently, current approaches to how manufacturing systems respond to changes in the business environment only focus on machine flexibility and system layout flexibility. However, these approaches lack the applicability to manufacturing systems with existing resources and also each approach is merely able to deal with one kind of change in the business environment. In the manufacturing industry, every change in the business environment can be directly or indirectly detected from customer demand. Hence, the most effective approach to helping manufacturing systems with existing resources deal with changes in the business environment is through operational flexibility, regardless of the need for new machine functionalities. This requires an effective production planning and scheduling method by which a manufacturing system is able to cost-effectively

fulfil customer demand in order to sustain their business competitiveness. More importantly, the planning and scheduling method must also be able to flexibly organise and utilise available resources to satisfy customer demand related to changes in the business environment. A manufacturing system is always hierarchical, wherein resources are grouped into sub-systems in line with given product design and process design through considering cost and time efficiency in production. Therefore, to satisfy the aforementioned cost-efficient planning and scheduling requirement, this method must enable customer demand to be fulfilled within existing system structure as far as possible. Nevertheless, if that is not possible due to dynamic changes in the business environment, this method must then enable resources to be flexibly regrouped across boundaries between different sub-systems for satisfying customer demand and dealing with the changes. Ultimately, a prerequisite for the proposed production planning and scheduling method is that there must be a manufacturing system control model that is able to represent the hierarchy of manufacturing system.

Control models for manufacturing systems are usually based on implementation of three typical architectures: centralised architecture (Dilts et al., 1991), hierarchical architecture (Jones and McLean, 1986; Jackson and Jones, 1987) and heterarchical architecture (Lin and James, 1992; Gu et al., 1997; Macchiaroli and Riemma, 2002; Wong et al., 2006a). In centralized and hierarchical architecture there is a singular centralized controller, or a hierarchy of many, that is/are rigidly designed based on static system status and global objectives of the manufacturing system. These two architectures are able to represent the physical hierarchy of a manufacturing system and also offer advantages such as global optimisation, predictability, and robustness in production planning and scheduling (Dilts et al., 1991; Chiu and Yih, 1995). Despite this, they lack operational flexibility due to centralized control. Conversely, the heterarchical architecture contains a collection of local controllers for individual resources within a manufacturing system without the existence of a hierarchy. Local controllers are given full autonomy to make local decisions based on their local status and objectives, whereas global decisions are made through the interactions amongst local controllers (Heragu et al., 2002). Due to the distributed nature of the architecture, heterarchical control models are very flexible and fault-tolerant. Also, resources within a manufacturing system can be self-organised together without centralised control, which produces real operational flexibility in decision-making. However, because of lack of hierarchy, applications of heterarchical architecture merely concern resource allocation in shop floor or single layer systems, and are unable to address the production planning and scheduling within manufacturing system having a complex hierarchy. In addition, since each local controller attempts to achieve its local objectives without considering global objectives, global control decisions based on heterarchical architecture are not always optimised. Therefore, none of the aforementioned typical architectures are able to simultaneously satisfy the prerequisite of a control model defined previously and at the same time provide production planning and scheduling methods with both features of cost-efficiency and flexibility. A new modelling and control architecture with the following three generalised features is therefore

needed:

- This architecture must be able to represent the complex physical hierarchy of manufacturing system.
- This architecture must avoid centralized control so as to achieve operational flexibility of manufacturing systems.
- This architecture must have a planning and scheduling method that is able to cost-efficiently fulfil customer demand within the structural constraints of the manufacturing system and also have the ability to flexibly regroup resources across system boundaries when needed. At the same time, because resource regroupings across system boundaries will create disturbances to manufacturing system, this method must be able to find the resource regrouping with the lowest disturbances.

In 2007, Zhang et al. (2007) proposed a manufacturing system methodology that is termed as Dynamically Integrated Manufacturing System (DIMS). This methodology encompasses a multi-agent modelling architecture – Hierarchical Autonomous Agent Network (HAAN) and an agent-based control method – Hierarchical Agent Bidding Mechanism that is particularly designed for production planning and scheduling in Make-to-Order (MTO) manufacturing system. This control method enables the hierarchy of resources in a MTO manufacturing system to not only be automatically controlled but also to be cost-efficient and flexibly self-organised within structural constraints and across system boundaries when needed. DIMS perfectly addresses the three features mentioned above. However, in this work, Zhang et al. (2007) did not demonstrate the effect of the bidding mechanism on the operational flexibility of a manufacturing system, nor its ability to deal with changes in the business environment. As a following work, this paper aims to fulfil the demonstration work that was not covered in Zhang et al. (2007). As a reminder, this paper is organised into six sections. Following the introduction section, section 2 provides a literature review of the emerging modelling architectures aside from the three typical architectures. Section 3 introduces the modelling architecture in DIMS. Section 4 presents the hierarchical agent bidding mechanism in DIMS and illustrates the hierarchical planning and scheduling process based on this mechanism. Section 5 demonstrate the positive effect of the agent bidding mechanism on operational flexibility of manufacturing system and its ability to help manufacturing system deal with changes in the business environment, especially change in demand quantity, on the basis of an industrial example. Section 6 is a conclusion section.

2. Modelling and Control Architectures of Manufacturing System

As mentioned in the previous section, centralised and hierarchical architectures offer the ability to model system hierarchy and globally optimise control decisions, but do not support the flexible operational decision-making. On the contrary, heterarchical architecture facilitates manufacturing systems to make

operational decisions flexibly, but cannot represent system hierarchy and achieve global optimisation in decision-making. With a view to combining the positive features of centralised/hierarchical and heterarchical architectures, hybrid architecture has attracted much academic attention over the past two decades and several approaches have been developed for constructing hybrid control architecture in manufacturing system.

One of the simplest approaches to hybrid architecture is to involve a level of global control over the coordination between a set of heterarchical resources. This global control is usually provided through introducing a supervisory controller into heterarchical architecture. In most cases, the supervisory controller is responsible for managing and controlling the coordination amongst heterarchical resources so that global objectives/decisions can be achieved/optimised (Butler and Ohtsubo, 1992; Ou-Yang and Lin, 1998; Mes et al., 2007; Li et al., 2010). Some other cases use a supervisory controller to achieve interfacing of process planning and production scheduling, where process plans are retrieved or created by supervisory controllers based on centralised methods. Production schedules in these cases are then obtained through heterarchical control on the basis of the process plan created by the supervisory controller (Wong et al., 2006b). The hybrid architecture based on this approach is easy to understand and implement. However, one of the significant shortcomings of this architecture is that it has the same downfalls as the heterarchical architecture in terms of system modelling; that is to say, it is only able to model single-level systems and cannot be used for modelling and controlling manufacturing systems with complex hierarchies.

Recently, two alternative approaches to hybrid architecture, in which manufacturing systems are represented with a hierarchy of heterarchical resources, have attracted much scholarly attention. These two approaches are based on two manufacturing system concepts: Holonic Manufacturing System (HMS) (Van Brussel et al., 1998) and Fractal Manufacturing System (FrMS) (Ryu and Jung, 2003; Ryu et al., 2003). In HMS and FrMS, each heterarchical resource is modelled and controlled by a generic unit referred to as a 'holon' or a 'fractal' in HMS and FrMS respectively. A generic unit is able to contain a hierarchy of children units, as well as in itself being part of a larger unit at higher level. Therefore, hybrid architectures based on HMS and FrMS are able to represent complex hierarchies of manufacturing systems and simultaneously offer the operational control flexibility that comes with the heterarchical feature of the modelling unit. In HMS, most of the applications are based on a typical architecture referred to as the Product-Resource-Order-Staff reference Architecture (PROSA) (Van Brussel et al., 1998). This architecture consists of three kinds of basic holons (generic units) that are used to model products, resources, and orders in respective manufacturing system. In addition, there is another reference holon in PROSA, termed as 'staff holon' that offers a level of centralised control over the heterarchical operations of basic holons. This staff holon assists the basic holons in performing their tasks within the consideration of the global objectives of the entire system. Van Brussel et al. (1998) argued that the concurrent presence of basic holon and staff holon means HMS has the inherent

ability to allow for the combination of heterarchical and hierarchical controls which in turn facilitates the generation of robust and flexible control decisions. However, most applications of HMS focus on dynamic scheduling in bottom-level sub-systems of manufacturing system (Gou et al., 1998; Sousa and Ramos, 1999; Bongaerts et al., 2000; Babiceanu et al., 2004; Babiceanu and Chen, 2007; Leitão and Restivo, 2006) or real-time production control and rescheduling in manufacturing systems (Cheng et al., 2004; Hsieh, 2004; Colombo et al., 2006; Blanc et al., 2008; Leitão and Restivo, 2008; Verstraete et al., 2008; Borangiu et al., 2009). There are only a few incidents of applying HMS to advanced production planning and scheduling within manufacturing systems with two or more hierarchical layers, such as the work of Gou et al. (1998). However, these applications are unable to deliver flexible resource regroupings across the structural constraints between various manufacturing sub-systems. For example, Gou et al. (1998) presented a holonic control architecture for production scheduling in a factory consisting of several cells. In their work, every product in the factory contained several sequenced parts, each of which could, however, be only produced within a specified cell.

In FrMS, there are very few applications concerned with planning and scheduling in manufacturing systems. Based on the original concept of FrMS, (Shin et al., 2009b,a) developed a 'Relation-driven Fractal Manufacturing System' (r-FrMS) to help MTO manufacturing systems hierarchically plan and schedule customer orders. In r-FrMS, manufacturing resources are modelled as Autonomous and Intelligent Resource (AIR) units (i.e., fractals) which are organised hierarchically through the employer-employee relationships. When r-FrMS receives a product order, the AIR unit representing the entire system will announce production jobs for different product parts to its employees (e.g., manufacturing cells). The negotiation between an employer and its employees is based on the Contract Net Protocol (CNP) mechanism. As for each employee AIR unit, in order to bid for a production job released from the system AIR unit, it in turn releases the sub-jobs of this production job to its own employees. This process is hierarchically carried out from the AIR unit at the top level to the AIR units at the bottom level after which operational decisions are generated and integrated gradually from the bottom level to the top level. The r-FrMS provides a planning and scheduling method that enables product orders to be fulfilled by dynamic resource groupings through coordination amongst a hierarchy of resources. However, since the coordination amongst AIR units is simply based on CNP, it cannot produce globally optimised control decisions. Moreover, r-FrMS does not support the relaxation of structural boundaries of manufacturing system and therefore cannot enable manufacturing resources to be regrouped flexibly across system boundaries.

From the above literature review regarding hybrid architecture, it is revealed that some hybrid architectures are able to represent complex hierarchies of manufacturing systems whilst avoiding (FrMS) or partially avoiding (HMS) centralised control in decision-making. However, within these architectures, there is not an effective production planning and scheduling method that enables a hierarchy of resources in a manufacturing system to be self-organised

cost-efficiently within the structural boundaries of the manufacturing system, and simultaneously to be regrouped flexibly across the boundaries in response to dynamic changes in the business environment. This then demonstrates the significance and necessity of the HAAN architecture and the hierarchical agent bidding mechanism proposed by Zhang et al. (2007).

3. Hierarchical Autonomous Agent Network

In DIMS, both manufacturing system and product order are represented as HAANs composed of a hierarchy of autonomous agents. The HAAN of a manufacturing system is created through two processes: physical decomposition of manufacturing system and agent registration. In the first process, the manufacturing system is decomposed into a hierarchy of manufacturing resources according to its organisational structure. Every manufacturing resource, as well as the entire system, is then modelled as an individual agent that is given crucial information about its physical entity such as hierarchical position and functionalities. For example, considering a shop floor (S) containing two cells (C1 and C2) each of which involves two machines (C1: M11 and M12, C2: M21 and M22), seven agents are created in the physical decomposition process, including a shop floor agent (SA), two cell agents (CA1 and CA2) and four machine agents (MA11, MA12, MA21 and MA22). In the second process, agents representing manufacturing resources at lower hierarchical levels register as children in particular agents representing manufacturing resources at higher levels. As in the above example, the two cell agents (CA1 and CA2) register in the shop floor agent (SA) as children agents. In a recursive way, the four machine agents will register as children agents in specific cell agents, i.e., MA11 and MA12 register in CA1 and MA21 and MA22 register in CA2. A system HAAN formed through the two processes represents the hierarchical structure of manufacturing system. Moreover, the simple connections between agents enable the HAAN's structure to be dynamically changed as agents register with or deregister from their parents.

Product orders are modelled as HAANs through a similar approach to modelling manufacturing system. Agents in this sense represent the whole product orders and production jobs of components/operations, which are connected together through the registration process in line with their hierarchical relationships. In the modelling architecture of DIMS, agents in HAANs of product orders are generally termed as part agents, whereas agents in HAANs of manufacturing systems are referred to as resource agents.

4. Agent-based Hierarchical Production Planning and Scheduling

When a product order is received by a manufacturing system, the HAAN of the product order is built correspondingly. The part agents among this HAAN will then negotiate with the resource agents in the HAAN of the manufacturing system for planning and scheduling this order, as in Figure1. The negotiation

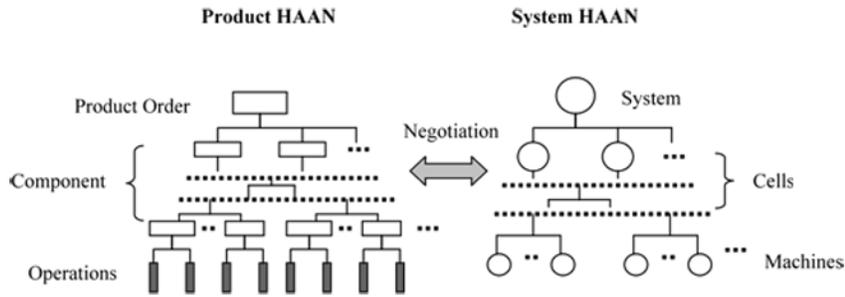


Figure 1: HAANs of manufacturing system and product order

between the HAAN of a product order and the HAAN of a manufacturing system, is based on a bidding approach. In this approach, a product order is gradually decomposed into production jobs along the order HAAN's hierarchy from top to bottom. At the same time, calls for bids of the production jobs are created by their part agents, which are then released to resource agents in the HAAN of manufacturing systems to bid for. In DIMS, a hierarchical agent bidding mechanism is developed, which is followed by every resource agent in the HAAN of a manufacturing system to process a call for bid received and make local decisions of planning and scheduling the production job related to this call for bid. The generic rules of this mechanism include that:

- R1. When planning and scheduling a production job in a manufacturing system, this mechanism gives priorities to those sub-systems that can fulfil the entire job within their structural constraints. For example, if a production job can be completed entirely within a sub-system, production plans and schedules where the job is entirely allocated to a specific sub-system are preferred over others in which resources across different sub-systems are grouped together. This is to ensure that the product/part will be manufactured cost-efficiently through following the economical manufacturing process inherent to the existing system structure.
- R2. If there are several sub-systems which are all technically able to process the entire production job, this mechanism then lets these sub-systems compete with each other and select the one offering the lowest production cost and satisfying the job's due date.
- R3. If any sub-systems cannot process the production job entirely but are able to process one or more sub-job, this mechanism will then attempt to find a collective production plan in which every sub-job is entirely allocated to one sub-system.
- R4. If the production job still cannot be completed satisfactorily by allocating sub-systems to sub-jobs, this mechanism will gradually relax the boundaries between sub-systems so that resources across sub-systems can be regrouped flexibly for fulfilling the job.

The generic rules described above, are to ensure that: every resource agent considers the structural constraints of the physical manufacturing entity represented, in production planning and scheduling of production jobs. As far as possible, production jobs are fulfilled within the structural constraints. However if this is not possible, the production jobs could be alternatively completed with flexible resource regrouping across the structural constraints when needed, but with the minimum disturbances. In the generic rules, R3 evokes a heterarchical resource allocation problem that is related to allocating a set of production jobs to a collection of resources. Therefore, in order to optimise the global planning and scheduling decisions, with respect to an entire product order, it is imperative to ensure that every heterarchical resource allocation problem incurred within the entire bidding process is solved optimally. To achieve this, Zhang et al. (2007) also developed an iterative agent bidding mechanism based on Genetic Algorithm (GA), which is embedded in the hierarchical agent bidding mechanism. Consequently, to demonstrate the hierarchical agent bidding mechanism clearly, the following section is organised into two subsections where the iterative agent bidding mechanism is illustrated first in section 4.1. Section 4.2 describes the general procedure of the hierarchical agent bidding mechanism.

4.1. Iterative Agent Bidding Mechanism

This paper uses a shop floor resource allocation problem for describing the iterative agent bidding mechanism. The problem in this context is concerned with finding the optimal allocation of a production job containing n operations (O_1, O_2, \dots, O_n) to a system consisting of m resources (r_1, r_2, \dots, r_m) so that the job can be processed completely before a due date D at the minimum production cost. It is assumed that these operations are to be completed one at a time according to a linear sequence and every resource contained in the system is able to process more than one operation if possible. Meanwhile, for every resource, there is a job buffer containing a list of unprocessed jobs from previous orders and these jobs are sorted in the buffer according to a time order. For each operation O_i ($i = 1, 2, \dots, n$), there is a subset of resources R_k ($k \leq m$) which are technically able to carry out this operation. As for each resource r_j ($r_j \in R_k$), it could place the operation at a specific position of its job buffer, which results in a scheduling option with a production cost ($C_{i,j}$) and a production lead time ($T_{i,j}$) represented as the following two equations, namely equation (1) and (2)

$$C_{i,j} = C_{i,j}^T + C_{i,j}^S + C_{i,j}^P + C_{i,j}^W + C_{i,j}^R \quad (1)$$

$$T_{i,j} = T_{i,j}^T + T_{i,j}^S + T_{i,j}^P + T_{i,j}^W \quad (2)$$

where

$C_{i,j}^T$ and $T_{i,j}^T$ = the transportation cost and time that depend on the location of the resource processing the previous operation (O_{i-1}), the location of current scheduling resource (r_j) and the material handling system.

$C_{i,j}^S$ and $T_{i,j}^S$ = the setup cost and time that depend on the previous unprocessed job in the job buffer of current scheduling resource (r_j).

$C_{i,j}^P$ and $T_{i,j}^P$ = the processing cost and time to complete the operation O_i on resource r_j .

$C_{i,j}^W$ and $T_{i,j}^W$ = the waiting/holding cost and time which depend on how many unprocessed jobs are placed before the operation O_i in the job buffer of current scheduling resource (r_j) how long it takes for r_j to process these unprocessed jobs.

$C_{i,j}^R$ = the rescheduling cost which is incurred when the operation O_i is placed before some unprocessed jobs already existing in the job buffer because this may result in additional costs to process these unprocessed jobs. However, the rescheduling must ensure the due dates of these unprocessed jobs are not violated.

The resource allocation problem described above then can be formulated as following in (3), where C and T respectively represent the overall production cost and lead time for processing the entire production job, while C_i and T_i are the production cost and lead time for every individual operation O_i ($i = 1, 2, \dots, n$).

$$\begin{aligned} \text{Min}(C &= \sum_{i=1}^n C_i) \\ T &= \sum_{i=1}^n T_i \leq D \end{aligned} \tag{3}$$

As it is broadly acknowledged that the number of possible allocation plans for a resource allocation problem is increased exponentially with the number of operations, solving such a problem by considering all possible plans in an enumerative way is unrealistic. To solve this, the iterative agent bidding mechanism uses an evolutionary optimisation process based on GA to find the optimal allocation plan. In this mechanism, two sets of parameters, a set of virtual prices (P_1, P_2, \dots, P_n) for operations and a set of minimal virtual profits ($F_{min,1}, F_{min,2}, \dots, F_{min,m}$) for resources, are introduced to control the behaviours of individual resource agents in the bidding process and in turn to create alternative allocation plans throughout different bidding iterations. Among the entire domain of feasible allocation plans, the optimum plan is located at the sub-domain that includes those satisfying the due date of the entire production job. As a result, to find the optimum allocation plan, it is imperative to ensure that the allocation plans obtained through the bidding iterations should firstly meet the due date of the entire production job. Therefore, the bidding criterion for selecting operation resources in the iterative bidding process is the shortest lead-time, so that the allocation plan obtained at a bidding iteration can have high possibility of meeting the due date of the entire production job. The optimum plan, with due date satisfaction and lowest production cost, is then found over the course of multiple bidding iterations by adjusting the virtual prices for operations and minimal virtual profits for resources.

To solve the resource allocation problem defined above, the production job is modelled by a part agent whilst the system and the m resources are represented by resource agents, based on the modelling architecture of DIMS. However, in order to ease the description of how the iterative agent bidding mechanism is used for the problem-solving, the part agent representing the entire production job is specifically termed as job agent. Also, the resource agent representing the entire shop floor system is particularly referred to as system agent, in order to explicitly differentiate it from resource agents representing the bottom resources in the shop floor.

The iterative agent bidding process is started by the job agent announcing a call for bids from the system agent which then runs the bidding process iteratively, as shown in Figure 2. Within each bidding iteration, a set of virtual prices (P_1, P_2, \dots, P_n) and a set of minimal virtual profits $(F_{min,1}, F_{min,2}, \dots, F_{min,m})$ are created first. $P_i (i = 1, 2, \dots, n)$ represents the virtual price assigned to operation O_i and $F_{min,j} (j = 1, 2, \dots, m)$ is the minimal virtual profit assigned to resource r_j . The system agent then announces operations, one at a time according to the specific sequence between operations, for resource agents to bid for. For each operation $O_i (i = 1, 2, \dots, n)$, when resource $r_j (r_j \in R_k)$, is technically able to process this operation, its resource agent will make a scheduling option for the operation and calculate the corresponding production cost $(C_{i,j})$ and lead time $(T_{i,j})$ based on the equation of (1) and (2). Subsequently, a virtual profit $(F_{i,j})$ earned by the resource to carry out this scheduling option is computed as the margin between the virtual price P_i given to operation and the production cost corresponding to the scheduling option, as shown in equation (4).

$$F_{i,j} = P_i - C_{i,j} \quad (4)$$

If based on the scheduling option, resource r_j is able to make a virtual profit greater than or equal to the minimal virtual profit assigned to it (i.e., $F_{min,j}$), its resource agent will submit the scheduling option as a bid to the system agent. In addition, since the rescheduling is allowed, resource r_j is technically able to have several scheduling options for operation O_i by placing the operation in different buffer positions without violating the due dates of the unprocessed jobs existing in the buffer. As long as these options satisfy the minimal virtual profit condition, they can be all submitted as bids to the system agent. Therefore, a resource agent may simultaneously submit multiple bids for one operation within a bidding iteration. When the system agent receives more than one bid for operation O_i from resource agents, it selects the bid corresponding to the scheduling option that has the shortest lead-time as the winner bid in this iteration, as shown in (5)

$$\begin{aligned} B_i^{win} &= B_i^{(l)}, T_i^{win} = T_i^{(l)}, C_i^{win} = C_i^{(l)} \\ T_i^{(l)} &= \text{Min}(T_i^{(1)}, T_i^{(2)}, \dots, T_i^{(G)}) \end{aligned} \quad (5)$$

where B_i^{win} , T_i^{win} and C_i^{win} represent the winner bid for operation O_i and the production lead time and cost corresponding to the winner bid, whereas G is the

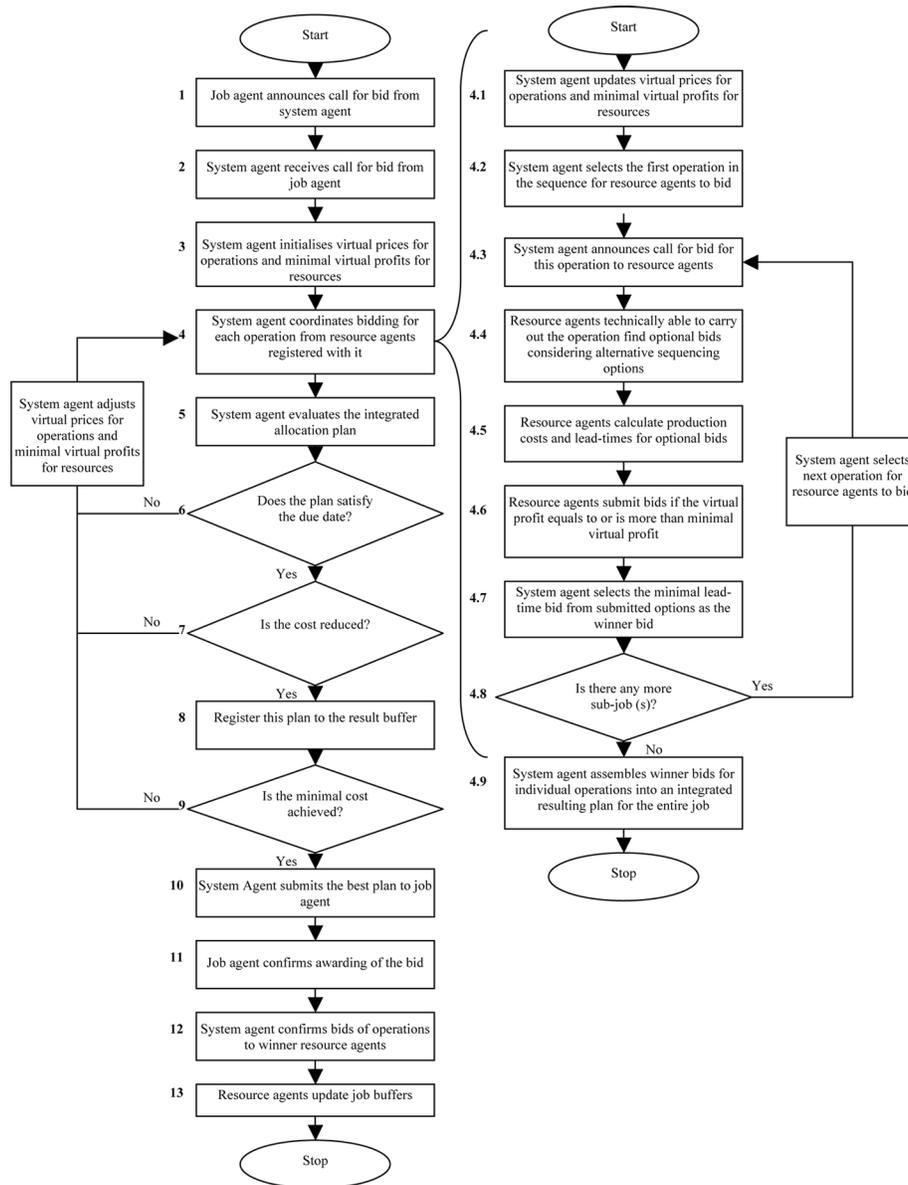


Figure 2: The Iterative Agent Bidding Process

total number of bids for operation O_i received by the system agent. However, if there is not a bid for operation O_i , it means the system agent cannot find an appropriate resource for O_i under the given conditions (i.e., virtual prices for the operations and minimal virtual profits for resources) and obviously cannot find an allocation plan for the entire production job in the current bidding iteration. Under such circumstances, the system agent terminates the current bidding iteration immediately and starts the next bidding iteration. On the other hand, when winning bids for all operations in the current iteration are found, the system agent then integrates these winner bids to form an allocation plan and calculates the planned cost (C) and lead time (T) for the entire production job as in (6)

$$C = \sum_{i=1}^n C_i^{win}, T = \sum_{i=1}^n T_i^{win} \quad (6)$$

If the planned lead time does not satisfy the required due date for the entire job, i.e., $T > D$, the system agent launches the next bidding iteration with adjusted virtual prices for operations and minimal virtual profits for resources. However, if the due date is satisfied, the system agent investigates whether the cost is reduced compared the last iteration. If so, the system agent places the generated plan into a result buffer before the next iteration is started. Otherwise, next bidding iteration is started subsequently. The iterative bidding process is ceased when the system agent considers the best plan in the result buffer is able to achieve minimal or sufficiently low cost.

In the iterative agent bidding process, different virtual prices for operations and minimal virtual profits for resources can lead to different allocation plans. Specifically, the higher virtual prices for operations or lower minimal virtual profits for resources encourage eligible resources to put forward more bids for operations (e.g., by the rescheduling process) so as to find allocation plans with short lead time. However, the lower virtual prices for operations or higher minimal virtual profits for resources reduce the attractiveness of operations for resources to submit high-cost/short-time bids and in turn lead to allocation plans with low production costs. In this context, when a set of virtual prices and a set of minimal virtual profits cannot result in an allocation plan meeting the due date, it implies that virtual prices for operations should be increased or minimal virtual profits for resources should be reduced so as to encourage resources to submit more bids. On the other hand, if the allocation plan in a bidding iteration is able to satisfy the due date, lower cost allocation plans could be found through reducing virtual prices for operations or increasing minimal virtual profits for resources in line with the two sets of parameters related to this plan.

As mentioned previously, a GA-based optimisation (as shown in Figure 3) is implemented to tune the virtual prices for operations and the minimal virtual profits for resources throughout bidding iterations so as to find the optimum allocation plan. A chromosome in this GA optimisation is a vector composed of a set of virtual prices for operations and a set of minimal virtual profits for

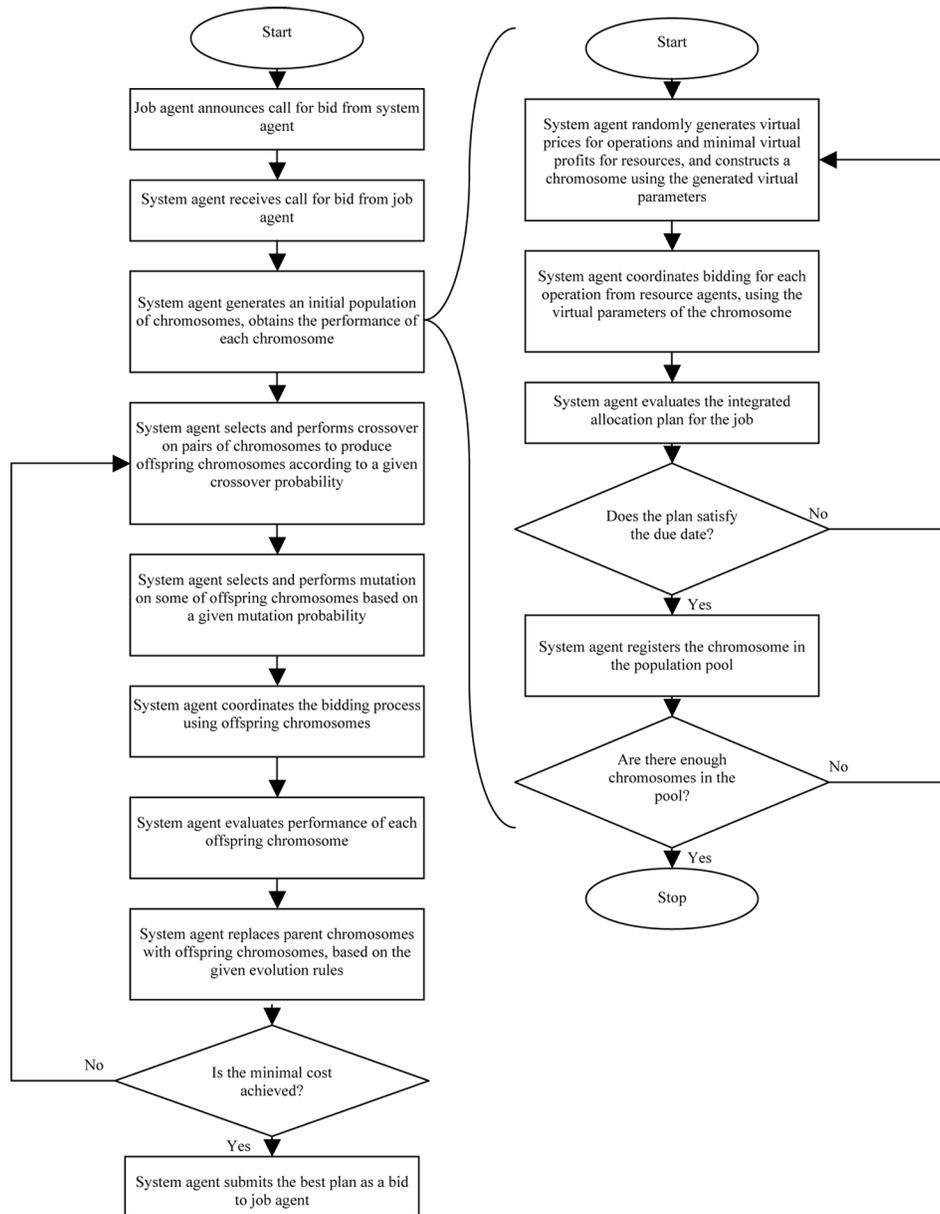


Figure 3: The GA-based Optimisation Process

resources, as in (7).

$$CH = \{P_1, P_2, \dots, P_n; F_{min,1}, F_{min,2}, \dots, F_{min,m}\} \quad (7)$$

These two sets of parameters contained in a chromosome, are used in the iterative agent bidding process to generate an allocation plan. Subsequently, the fitness function of the chromosome is represented by the production cost corresponding to the allocation plan. Low production costs represents high fitness and vice versa. When the system agent starts the iterative agent bidding process, the GA-based optimisation is simultaneously launched with the generation of an initial population of chromosomes. In the population initialisation, chromosomes are generated through a random process where the virtual prices for operations and the minimal virtual profits for resources are random values in a range of $(0, N)$. The number of N could be any value greater than the total cost of any operation processed by any eligible resource. Once a chromosome is created for population initialisation, the two sets of parameters of this chromosome are immediately used in the iterative agent bidding process, so as to investigate whether the chromosome is able to result in a feasible allocation plan (i.e., the planned lead time meets the due date). If the allocation plan meets the due date, this chromosome is then placed into the population pool. Otherwise, it will be discarded and the next initial chromosome is created subsequently through the random process. The chromosome initialisation is ceased once a predefined population size is reached. Based on the generated population, the GA-based optimisation can subsequently take place. Within each GA iteration, a set of offspring chromosomes are produced through the crossover and mutation processes. These offspring chromosomes are then used in the bidding process so as to find alternative allocation plans. For those offspring chromosomes that are able to result in feasible allocation plans, a fitness assessment takes place through comparing their fitness with that of their parents. Offspring chromosomes with better fitness will be used to replace their parents in the population, which in turn will forms a new generation of chromosomes at the end of the GA iteration. The process of the GA-based optimisation iterates until the system agent ceases the iterative agent bidding process.

4.2. Hierarchical Agent Bidding Mechanism

Considering a general bidding scenario: a Resource Agent (RA) receives a call for bid for a production job represented by a Part Agent (PA), the generic procedure of the hierarchical agent bidding mechanism is then described in the five steps below and in Figure 4.

S1: RA first checks whether the manufacturing system represented has sub-systems so as to subsequently find one sub-system for optimally processing the entire production job. To check this, RA will investigate its internal HAAN structure that represents the physical structure of the manufacturing system.

If RA has no registered children agents, it is implied that RA represents a resource at the bottom hierarchical level of the manufacturing system e.g.,

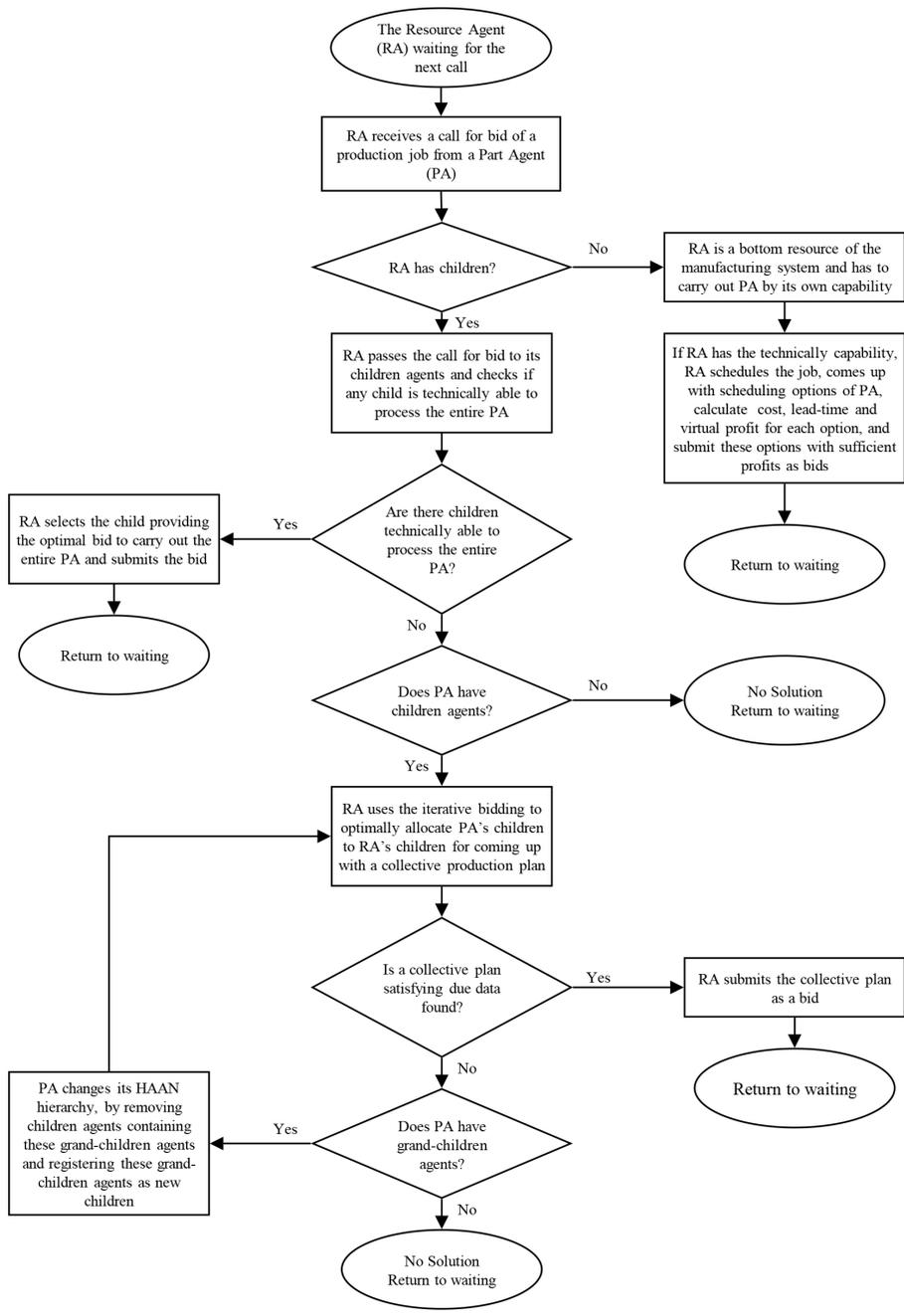


Figure 4: The Generic Procedures of Hierarchical Agent Bidding Mechanism

a machine. Hence, the job has to be entirely completed by RA itself. Thus, if RA is technically able to perform the production job, it will schedule the job based on its own capability and then submits the obtained schedules as bids to PA, as long as the bids satisfy conditions of the call for bid e.g., due date. Ultimately, no matter whether RA successfully submits bids to PA, it returns to the waiting status and the bidding process for the current call for bid stops.

Else, go to step 2.

- S2: The manufacturing system represented by RA has sub-systems. Therefore, RA will investigate which sub-system is the optimal one to complete the entire production job. To achieve this, RA hands over the call for bid directly to its children agents and waits for their bidding responses.

If RA receives one or more bids from some of its children agents, it will select the optimal bid and submit it to PA. RA then returns to the waiting status and the bidding process for this call for bid stops. In this case, if the production job is for an entire product order, the lowest production cost is used as the criterion for selecting the optimal bid. Otherwise, the shortest lead-time will be used for optimal bid determination.

Else, go to step 3.

- S3: RA does not receive any bids from its children agents, which implies that no one sub-system is technically able to complete the entire job and concurrently satisfy the conditions of the call for bid. Therefore, to come up with a production plan under the given conditions, different sub-systems are to be grouped collectively, each of which will complete one or more parts of the production job. In this sense, the prerequisite for creating a collective plan, namely if the production job can be decomposed into children jobs, has to be checked first.

If PA has no children agents, it means the production job is at the bottom level of the product hierarchy, and in turn the job cannot be collectively processed across different sub-systems. Coupled with the failure in allocating the job to a singular sub-system, it is implied that the production job cannot be completed within the system represented by RA. RA then returns to the waiting status and the bidding process for the call for bid stops.

Else, go to step 4.

- S4: PA has children agents, which denotes that the production job is able to be collectively produced. In order to lead minimum disturbances to the existing system structure as proposed, RA has to firstly consider such a collective plan: the production job is collectively completed by multiple sub-systems, whereas every child job is entirely processed within a singular sub-system. Thus, RA faces a resource allocation problem, namely how to find the optimum plan of allocating a set of children jobs to a set of sub-systems. To solve this problem, the iterative agent bidding mechanism illustrated previously, is used. Within each bidding iteration, RA will announce calls for bids regarding children jobs, one at a time, for its children agents to bid upon.

If RA finds a collective plan that is able to meet the bidding conditions of the call for bid regarding the entire job, it will submit the plan as a bid to PA. RA then returns to the waiting status and the bidding process for the call for bid regarding the entire job stops.

Else, go to step 5.

S5: When RA is unable to submit a bid in step 4, the constraint aforementioned in step 4 for finding a collective plan, namely every child job is entirely completed within a singular sub-system, is then released, so that children jobs can also be flexibly processed across sub-systems when needed. In doing so, the production job must be further decomposed and thus the hierarchies of children jobs need to be investigated again.

If PA has grandchildren agents, it is informed to remove children agents which contain these grandchildren agents from its hierarchy, and directly register all these grandchildren agents as new children. In this sense, the structure of the production job is changed, in which a new set of children jobs are created. The bidding process then goes back to step 4 to launch another iterative bidding process for finding a collective plan in which, the new set of children jobs are allocated to sub-systems.

Else, due to the failures in previous steps, it is concluded that the production job cannot be processed in the system represented by RA under the given conditions of call for bid regarding PA. RA then returns to the waiting status and the bidding process for the call for bid of PA stops.

When a product order is placed in a manufacturing system, the production planning and scheduling process based on the hierarchical agent bidding mechanism, is started from the part agent, which represents the entire order, calling for bids from the resource agent which represents the entire manufacturing system. This may result in the system agent directly passing the call for bids to its children agents, or the system agent gradually decomposing the order into a set of production jobs (e.g., children job, grandchildren jobs etc.), then announcing a call for bids for these jobs one at a time, for children agents to bid for and subsequently coordinating children agents based on the iterative agent bidding mechanism. Similar to the system agent, every child of the system agent also follows the hierarchical agent bidding mechanism to process calls for bids received. However, the bidding conditions in the call for bids (i.e., thresholds for resource agents to submit their bids) might be minimal virtual prices if the system agent starts an iterative bidding process for finding a collective production plan. In turn, the hierarchical agent bidding mechanism is used by every resource agent at lower levels to process calls for bids passed over from its parent agent. This leads to a hierarchical planning and scheduling process which involves a hierarchy of bidding processes operated respectively by the hierarchy of resource agents in the HAAN of the manufacturing system, all based on implementation of the hierarchical agent bidding mechanism. The bidding processes at lower levels are involved with those at higher levels based on parent-children relationships between corresponding resource agents. The control decisions in the hierarchical planning and scheduling process are made in a recursive way: the

decision-making incidents in bidding processes at higher levels rely on decision-making completion in those at lower levels. In this way, planning and scheduling decisions from the sub-systems/resources at lower levels are integrated and evaluated by those at higher levels, in line with both global objectives and dynamic situations in the manufacturing system. This helps the manufacturing system to reach both global optimisation and operational flexibility in fulfilling product orders.

During the implementation of the hierarchical agent bidding mechanism, most cases of resource regroupings occur in step 5 of the mechanism, namely when the hierarchical structure of the HAAN of a production job is changed. Here, a resource agent attempts to find the optimum allocation plan for the new generation of children of the production job. Once receiving a call for bid, the resource agent will firstly pass the call to their children at lower levels, because every resource agent follows the generic procedure of the mechanism when processing call for bids. Therefore, resource regroupings in the hierarchical planning and scheduling process first take place between sub-systems at lower levels (e.g., resource regroupings between cells of a shop), which are considered level by level from bottom to top along the system hierarchy until the production job is fulfilled. This enables resource regroupings to make minimum disturbances to the existing structure of manufacturing system.

5. Experimental Application

5.1. Test Case

This section presents an application of the hierarchical agent bidding mechanism at a Mexico manufacturing company. It aims to demonstrate the proposed abilities of this mechanism in cost-efficient planning and scheduling and flexible resource regroupings in response to dynamic changes in the business environment. In this application, dynamic changes in the business environment only refers to change in customer demand, i.e., demand fluctuation. Due to the confidential agreement, the company name is not shown in this paper.

Although the company has a very complex structure that consists of several workshops for producing different products or components, only two workshops (WA and WB) are selected to make the test in this paper. WA and WB are responsible for producing two products respectively, namely the Block Mold (BM) and the Head Mold (HM). The structures of WA and WB are shown in Figure 5 and Figure 6, whilst the structures of the two products are depicted in Figure 7 and Figure 8 respectively. The functionalities of machines and operations of products are given in Table 1. However, detailed operational parameters of machines and products are not shown in this paper due to the confidential agreement. Table 2 shows two process plans based on which BM and HM are produced in the company and the authors believe that both plans are cost-efficient as the two workshops are designed specifically for the two products.

Since the company were not willing to provide order information for reasons of protecting clientele, this paper creates 11 sets of orders (as in Table 3) of

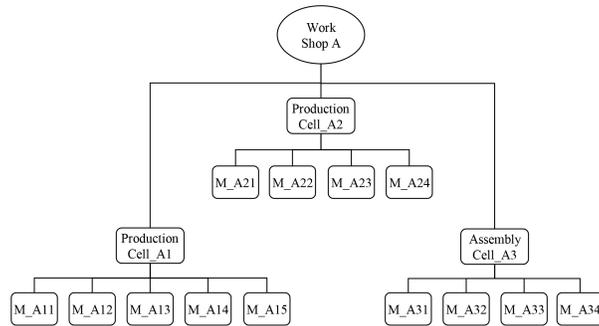


Figure 5: The Layout of Shop A

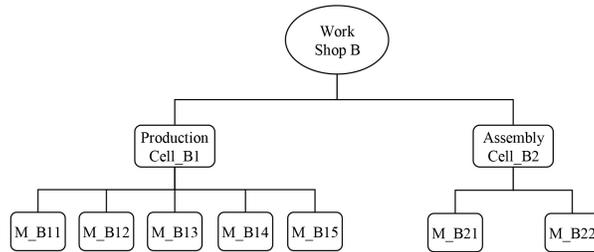


Figure 6: The Layout of Shop B

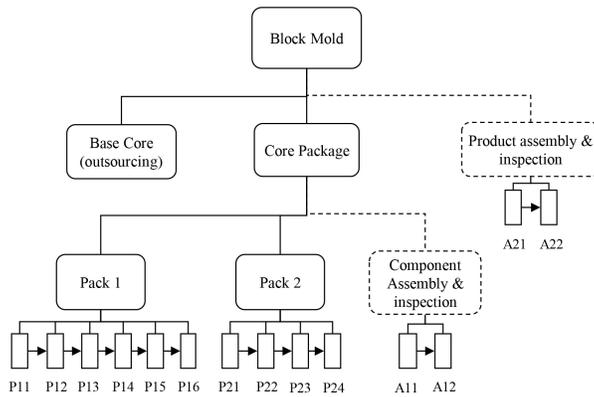


Figure 7: The Structure of BM

BM and HM. Each order set covers a number of orders over 5 simulation days. In the first set, the number of daily orders for BM and HM are 6 and 4 respectively, whereas each following set increases the number of daily orders for BM by 1 compared to that in the previous set. Orders on each simulation day are released in a random sequence and the rule for processing these orders is

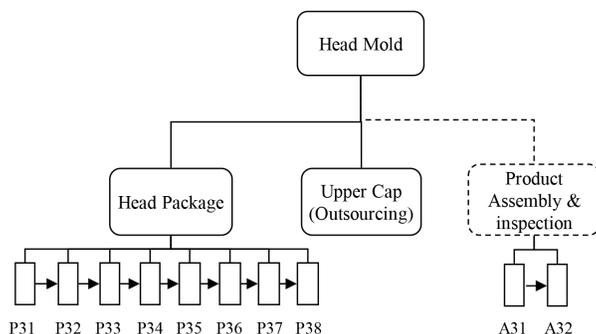


Figure 8: The Structure of HM

<i>Machine</i>	<i>Functionality</i>	<i>Operations</i>	<i>Required Functionality</i>
M_A11	f1, f2, f5	P11	f1
M_A12	f1, f2, f5	P12	f1
M_A13	f1, f2, f5	P13	f2
M_A14	f3, f4	P14	f2
M_A15	f3, f4	P15	f3
M_A21	f2, f5	P16	f4
M_A22	f2, f5	P21	f5
M_A23	f3, f4	P22	f5
M_A24	f3, f4	P23	f3
M_A31	f6	P24	f4
M_A32	f6, f8	A11	f6
M_A33	f7	A12	f7
M_A34	f7, f9	A21	f8
M_B11	f1, f2, f5	A22	f9
M_B12	f1, f2, f5	P31	f1
M_B13	f10	P32	f1
M_B14	f10	P33	f5
M_B15	f3, f4	P34	f5
M_B21	f6, f8	P35	f5
M_B22	f7, f9	P36	f10
		P37	f3
		P38	f4
		A31	f6
		A32	f7

Table 1: The Data about Machines and Product Operations

First Come First Served (FCFS). Every order for BM and HM is placed with a unique pattern in this test: batch sizes for BM and HM orders are 10 and 3 respectively and preferred lead-times (the maximum time gap between arrival time and delivery time) for every order of BM or HM is 2 simulation days. The batch size and lead-time conditions exist to facilitate the data analysis regarding systems performance when experiencing demand fluctuation.

<i>Shops/Cells</i>	<i>Products/Components or Assembly</i>
Shop A	Block Mold
Cell_A1	Pack1
Cell_A2	Pack2
Cell_A3	Assembly of Core Package
Cell_A3	Assembly of Block Mold
<hr/>	
Shop B	Head Mold
Cell_B1	Head Package
Cell_B2	Assembly of Head Mold

Table 2: The Cost-efficient Process Plans for BM and HM

<i>Order Set</i>	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th	10 th	11 th
<i>Number of daily orders of BM</i>	6	7	8	9	10	11	12	13	14	15	16
<i>Number of daily orders of HM</i>	4	4	4	4	4	4	4	4	4	4	4

Table 3: The Created Order Sets for the Experiment

5.2. Test Results and Analysis

Figure 9 depicts the GA-based optimisation process in the hierarchical agent bidding mechanism when searching for the optimum production plan and schedule for the first BM order in the first order set. It can be observed that, with the GA iterations proceeding, the overall production cost for this BM order is gradually decreased from over 16,000 to approximately 14,750. This demonstrates that, although the hierarchical agent bidding mechanism is a distributed method, it enables a hierarchy of manufacturing resources within a manufacturing system to be self-organised, with the result of cost-efficiently fulfilling customer orders.

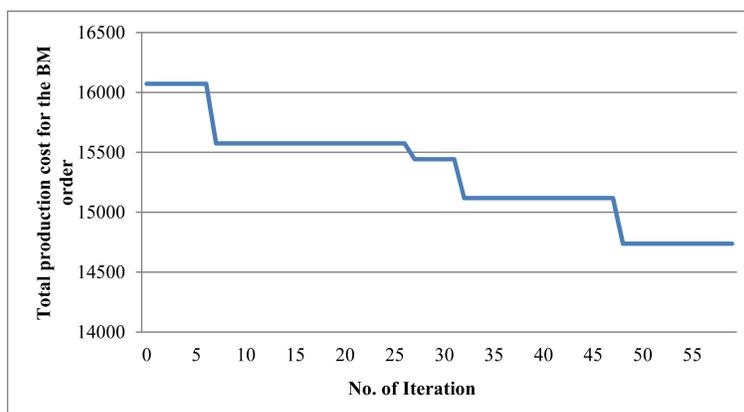


Figure 9: The GA-based Optimisation Process for Planning and Scheduling the 1st BM Order in the 1st Order Set

<i>Products</i>	<i>Number of daily Orders of BM</i>										
	6	7	8	9	10	11	12	13	14	15	16
BM made in Shop A	93%	91%	88%	87%	86%	84%	83%	85%	83%	81%	79%
Pack1 made in Cell_A1											
BM made in Shop A with such a plan:	93%	91%	80%	78%	74%	73%	72%	72%	71%	71%	66%
Pack2 made in Cell_A2											
Core Pack assembled in Cell_A3											
BM assembled in Cell_A3											
HM made in Shop B	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
HM made in Shop B with such a plan:	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Head Package made in Cell_B1											
HM assembled in Cell_B2											

Table 4: The Overall Production Planning and Scheduling Performance based on the Hierarchical Agent Bidding Mechanism

Table 4 shows the overall performance of the hierarchical agent bidding mechanism in planning and scheduling orders within all order sets. It is shown in this table that many BM orders, for example 93%, 91% and 88% in the first, second and third order sets, are completed entirely within WA. Meanwhile, most of BM orders, for example, 93%, 91% and 80% in the first, second and third order sets, are produced rigidly within the system structure of WA (i.e., every basic component of BM is completed by a single cell of WA). Also, the process plans produced by the bidding mechanism for these orders are consistent with the given cost-efficient plan shown in Table 2. Moreover, Table 4 shows that, with the increase of BM demand, the BM orders entirely produced within WA are gradually decreased from 93% to the 79%. This implies that, with BM demand increase, resources of WB are gradually utilised to produce more and more BM orders. Hence, the experiment results in Table 4 demonstrate two abilities of the hierarchical agent bidding mechanism: Firstly, it is able to consider the structural constraints of a manufacturing system in planning and scheduling customer orders and thus reap the potential benefits derived from the existing system structure e.g., low production cost, material handling cost etc. Secondly, when facing demand fluctuation, this mechanism is able to balance the production capacity across a complex manufacturing system and in turn enhance the systems operational flexibility to the demand fluctuation.

	<i>Number of daily Orders of BM</i>										
	6	7	8	9	10	11	12	13	14	15	16
BM made with resource regrouping across cell of Workshop A	0	0	3	4	6	6	7	8	8	8	10
BM made entirely within Workshop B	2	3	5	6	7	9	10	10	12	14	17

Table 5: No. of BM Orders Processed based on Resource Regroupings

Table 5 shows the number of BM orders that are completed by resource regroupings within WA (i.e. some basic components of BM are processed by multiple cells of WA together) and those processed entirely by WB throughout all order sets. It is observed that with the demand increase, BM orders that are processed based on resource regrouping within WA are increased from 0 in the first order set to 10 in the last order set. This validates the ability of the

hierarchical agent bidding mechanism to support resource regroupings across system boundaries and in turn verifies the capability of this mechanism to enhance the operational flexibility of manufacturing system to demand fluctuation. Meanwhile, it is indicated that throughout the eleven order sets, the BM orders processed entirely by WB are always more than the BM orders completed by resource regroupings within WA. This is because WB only has one production cell and thus, as long as a BM order is entirely completed within WB, it can be regarded that the order is processed within system structure of WB. Therefore, compared to the production plans and schedules evoking resource regroupings within WA, a plan and schedule by which a BM order is entirely produced within WB will result in fewer disturbances to the production system, which is given a higher priority in the hierarchical agent bidding mechanism. Theoretically, as long as WB has available capacity to manufacture BM orders, the redundant BM orders that cannot be fulfilled within the system structure of WA will be produced in WB but should not be produced with resource regroupings in WA. However, Table 5 indicates that, even though the BM demand is not very high (e.g., the first order set), resource regroupings within WA are still incurred sometimes. This is because the hierarchical agent bidding mechanism relies on a GA-based heuristic method to find optimal production plans and schedules and hence in some cases it might fail in finding plans and schedules within system structures due to particular reasons caused by random numbers.

In fact, if considering BM as a product newly introduced to the manufacturing system, the test results also demonstrate the ability of the hierarchical agent bidding mechanism to deal with dynamic change in product type, as it enables BM to be effectively fulfilled within WB without any given process plans.

6. Conclusion

This paper demonstrates the hierarchical agent bidding mechanism that is proposed in Zhang et al. (2007) for assisting MTO manufacturing systems in global optimised and flexible production planning and scheduling. Through an experimental test based on a Mexico manufacturing company, it is proved that: this mechanism is able to consider structural constraints of manufacturing systems in production planning and scheduling process, based on which, customer orders can be cost-efficiently fulfilled within structural constraints of manufacturing systems, whilst being able to be processed with flexible resource regroupings across structural boundaries but with minimum disturbances to the existing system structure when needed.

Although the hierarchical agent bidding mechanism was initially developed for MTO manufacturing systems, it is potentially applicable to alternative manufacturing systems such as Engineering-to-Order (ETO) systems, Assembly-to-Order (ATO) Systems, and even Make-to-Stock (MTS) systems. An ETO system consists of several departments, each of which has several teams. Projects in an ETO system are actually its products, which can be decomposed into a number of tasks, sub-tasks and so on. Therefore, an ETO system and its projects

can be modelled as HAANs that communicate with each other based on the hierarchical agent bidding mechanism for finding cost-efficient and flexible project plans and schedules. Similarly, an ATO system can use the HAAN architecture to model both its assembly system and assembly orders, and in turn use the hierarchical agent bidding mechanism to find cost-efficient and flexible operational decisions for assembly orders. In a MTS system, the hierarchical agent bidding mechanism can be integrated with the MRP system together. Specifically, the output from a MRP system could be regarded as an entire order that can be modelled as a product HAAN, negotiating with the manufacturing system HAAN based on the hierarchical agent bidding mechanism.

As concluded above, the hierarchical agent bidding mechanism is a promising way to assist manufacturing systems in making optimal and flexible operational control decisions in response to dynamic changes in the business environment. Despite this, the mechanism has one limitation, i.e., it is now only able to deal with orders one by one. It cannot currently be used to process multiple orders at the same time so as to globally optimise the overall performance of manufacturing systems. In order to effectively use the mechanism in the latter case, a method that is able to combine multiple orders together so as to create an order portfolio, is needed. The order portfolio will then be regarded as an entire order, which will then be planned and scheduled aggregately based on the use of the hierarchical agent bidding mechanism. The aforementioned method for integrating multiple orders is similar to MRP. However, the authors believe that, with the increase of differentiation between individual orders, especially orders in MTO or ETO systems, MRP is unable to solve the problem with respect to the integration of multiple orders. Therefore, this method must be more generic than MRP, so that it can be applicable to various scenarios. However, as a pre-work for this proposed future work, the application of the hierarchical agent bidding mechanism to MTS systems will firstly be investigated.

Babiceanu, R., Chen, F., 2007. Manufacturing scheduling in decentralised holonic systems using artificial intelligence techniques. *International journal of manufacturing technology and management* 11 (3), 389–410.

Babiceanu, R., Chen, F., Sturges, R., 2004. Framework for the control of automated material-handling systems using the holonic manufacturing approach. *International Journal of Production Research* 42 (17), 3551–3564.

Balakrishnan, J., Cheng, C. H., 1998. Dynamic layout algorithms: a state-of-the-art survey. *Omega* 26 (4), 507–521.

Balakrishnan, J., Jacobs, F. R., Venkataramanan, M. A., 1992. Solutions for the constrained dynamic facility layout problem. *European Journal of Operational Research* 57 (2), 280–286.

Benjaafar, S., Heragu, S. S., Irani, S. A., 2002. Next generation factory layouts: research challenges and recent progress. *Interfaces* 32 (6), 58–76.

- Blanc, P., Demongodin, I., Castagna, P., 2008. A holonic approach for manufacturing execution system design: An industrial application. *Engineering Applications of Artificial Intelligence* 21 (3), 315–330.
- Bongaerts, L., Monostori, L., McFarlane, D., Kádár, B., 2000. Hierarchy in distributed shop floor control. *Computers in Industry* 43 (2), 123–137.
- Borangiu, T., Gilbert, P., Ivanescu, N., Rosu, A., 2009. An implementing framework for holonic manufacturing control with multiple robot-vision stations. *Engineering Applications of Artificial Intelligence* 22 (4-5), 505–521.
- Butler, J., Ohtsubo, H., 1992. Addyms: architecture for distributed dynamic manufacturing scheduling. *Artificial Intelligence Applications in Manufacturing*, 199–214.
- Buzacott, J. A., Yao, D. D., 1986. Flexible manufacturing systems: a review of analytical models. *Management Science* 32 (7), 890–905.
- Cheng, F., Yang, H., Lin, J., 2004. Development of holonic information coordination systems with failure-recovery considerations. *IEEE Transactions on Automation Science and Engineering* 1 (1), 58–72.
- Chiu, C., Yih, Y., 1995. A learning-based methodology for dynamic scheduling in distributed manufacturing systems. *International Journal of Production Research* 33 (11), 3217–3232.
- Colombo, A., Schoop, R., Neubert, R., 2006. An agent-based intelligent control platform for industrial holonic manufacturing systems. *IEEE Transactions on Industrial Electronics* 53 (1), 322–337.
- Dilts, D., Boyd, N., Whorms, H., 1991. The evolution of control architectures for automated manufacturing systems. *Journal of manufacturing systems* 10 (1), 79–93.
- Gou, L., Luh, P., Kyoya, Y., 1998. Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation. *Computers in Industry* 37 (3), 213–231.
- Gu, P., Balasubramanian, S., Norrie, D., 1997. Bidding-based process planning and scheduling in a multi-agent system. *Computers & industrial engineering* 32 (2), 477–496.
- Heragu, S., Graves, R., Kim, B., St Onge, A., 2002. Intelligent agent based framework for manufacturing systems control. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 32 (5), 560–573.
- Hsieh, F., 2004. Model and control holonic manufacturing systems based on fusion of contract nets and petri nets. *Automatica* 40 (1), 51–57.
- Jackson, R., Jones, A., 1987. An architecture for decision making in the factory of the future. *Interfaces*, 15–28.

- Jones, A., McLean, C., 1986. A proposed hierarchical control model for automated manufacturing systems. *Journal of Manufacturing Systems* 5 (1), 15–25.
- Kochhar, J., Heragu, S., 1999. Facility layout design in a changing environment. *International Journal of Production Research* 37 (11), 2429–2446.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van Brussel, H., 1999. Reconfigurable manufacturing systems. *CIRP Annals-Manufacturing Technology* 48 (2), 527–540.
- Leitão, P., Restivo, F., 2006. Adacor: A holonic architecture for agile and adaptive manufacturing control. *Computers in industry* 57 (2), 121–130.
- Leitão, P., Restivo, F., 2008. A holonic approach to dynamic manufacturing scheduling. *Robotics and Computer-Integrated Manufacturing* 24 (5), 625–634.
- Li, X., Zhang, C., Gao, L., Li, W., Shao, X., 2010. An agent-based approach for integrated process planning and scheduling. *Expert Systems with Applications* 37 (2), 1256–1264.
- Lin, G., James, J., 1992. Integrated shop floor control using autonomous agents. *IIE transactions* 24 (3), 57–71.
- Macchiaroli, R., Riemma, S., 2002. A negotiation scheme for autonomous agents in job shop scheduling. *International Journal of Computer Integrated Manufacturing* 15 (3), 222–232.
- Mes, M., Van Der Heijden, M., Van Harten, A., 2007. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research* 181 (1), 59–75.
- Ou-Yang, C., Lin, J., 1998. The development of a hybrid hierarchical/heterarchical shop floor control system applying bidding method in job dispatching. *Robotics and Computer-Integrated Manufacturing* 14 (3), 199–217.
- Rosenblatt, M. J., HAU, L., 1987. A robustness approach to facilities design. *International Journal of Production Research* 25 (4), 479–486.
- Rosenblatt, M. J., Kropp, D. H., 1992. The single period stochastic plant layout problem. *IIE transactions* 24 (2), 169–176.
- Ryu, K., Jung, M., 2003. Agent-based fractal architecture and modelling for developing distributed manufacturing systems. *International Journal of Production Research* 41 (17), 4233–4255.
- Ryu, K., Son, Y., Jung, M., 2003. Modeling and specifications of dynamic agents in fractal manufacturing systems. *Computers in Industry* 52 (2), 161–182.

- Shin, M., Mun, J., Jung, M., 2009a. Self-evolution framework of manufacturing systems based on fractal organization. *Computers & Industrial Engineering* 56 (3), 1029–1039.
- Shin, M., Mun, J., Lee, K., Jung, M., 2009b. r-frms: a relation-driven fractal organisation for distributed manufacturing systems. *International Journal of Production Research* 47 (7), 1791–1814.
- Sousa, P., Ramos, C., 1999. A distributed architecture and negotiation protocol for scheduling in manufacturing systems. *Computers in Industry* 38 (2), 103–113.
- Stecke, K. E., 1983. Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Management Science* 29 (3), 273–288.
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: Prosa. *Computers in industry* 37 (3), 255–274.
- Verstraete, P., Valckenaers, P., Van Brussel, H., Saint Germain, B., Hadeli, K., Van Belle, J., 2008. Towards robust and efficient planning execution. *Engineering Applications of Artificial Intelligence* 21 (3), 304–314.
- Wong, T., Leung, C., Mak, K., Fung, R., 2006a. An agent-based negotiation approach to integrate process planning and scheduling. *International journal of production research* 44 (7), 1331–1351.
- Wong, T., Leung, C., Mak, K., Fung, R., 2006b. Dynamic shopfloor scheduling in multi-agent manufacturing systems. *Expert Systems with Applications* 31 (3), 486–494.
- Zhang, D., Anosike, A., Lim, M., 2007. Dynamically integrated manufacturing systems (dima) a multiagent approach. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 37 (5), 824–850.