B. TECH. PROJECT REPORT

On

Design and Development of Distributed Planning System For Smart Manufacturing

BY

Kartikeya Upasani and Miroojin Bakshi

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

November 2015

Design and Development of Distributed Planning System For Smart Manufacturing

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:
Kartikeya Upasani and Miroojin Bakshi

Guided by:
Dr. Bhupesh Kumar Lad

INDIAN INSTITUTE OF TECHNOLOGY INDORE
November 2015

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **'Design and Development of Distributed Planning System For Smart Manufacturing'** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering completed under the supervision of **Dr. Bhupesh Kumar Lad,** Assistant Professor, Mechanical Engineering, IIT Indore is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student(s) with date

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of BTP Guide(s) with dates and their designation

Preface

This report on "Design and Development of Distributed Planning System for Smart Manufacturing" is prepared under the guidance of **Dr. Bhupesh Kumar Lad**.

Through this report we have tried to present a system for smarter planning and decision-making in industries. The system is an effort towards making industries realize the potential of application of Cyber Physical Systems, Data Analytics and Computer Optimization to operations planning.

We have tried to the best of our abilities and knowledge to explain the content in a lucid manner.

Kartikeya Upasani and Miroojin Bakshi,
B.Tech. IV Year,
Discipline of Computer Science and Engineering,
IIT Indore.

Acknowledgements

We wish to thank Dr. Bhupesh Kumar Lad for his kind support and valuable guidance.

We wish to thank DST - Innovation and Entrepreneurship Development Centre (IEDC), and PRIUS IIT Indore for partial funding, and AVTEC Limited, Pithampur for an opportunity for industrial implementation, data collection and for support in kind.

The presented work was carried out at Intelligent Manufacturing Planning (IMP) Lab, IIT Indore, and we are thankful for the resources made available to us.

Without their support this report would not have been possible.

Kartikeya Upasani and Miroojin Bakshi
B.Tech. IV Year
Discipline of Computer Science and Engineering
IIT Indore

Abstract

With the rise of Internet of Things (IoT), the application of computer technology is becoming ubiquitous. The physical world is becoming an information system—through sensors embedded in physical objects and linked through wired and wireless networks. The project is on Smart Manufacturing, or what is considered to be the next Industrial Revolution- the application of Cyber-Physical Systems, Industrial IoT and intelligent algorithms to manufacturing enterprises for efficient planning and decision-making. Through this work, we aim to offer a low cost and industry ready software that can demonstrate the potential of smart manufacturing techniques.

<u>Table of Contents</u>

List of Figures

List of Tables

1. Introduction

With the rise of Internet of Things (IoT), the application of computer technology is becoming ubiquitous. The physical world is becoming an information system—through sensors embedded in physical objects and linked through wired and wireless networks. The perceivable potential of monitoring, automation and optimization is turning IoT from an application area into a research area. The manufacturing industry is no exception to the penetration of IoT, and is witnessing the advent of what is said to be the *Fourth Industrial Revolution*. Cyber-Physical Systems (CPS), data analytics and computer optimization are the main drivers of this change. The Fourth Industrial Revolution will be marked by a radical transformation in the modus operandi of enterprises. The role being played by human perception, intelligence and experience will be aided or even completely taken over by sophisticated sensing techniques and computer intelligence. The benefits of this include lesser reliance on human experience, more robust planning and efficient production.

Each asset of the enterprise (machines, departments) are brought online, onto a communication network. Enterprise data is collected and stored in a fashion that will make its scientific analysis possible. The status of all assets in the enterprise is monitored in real-time. Whenever an event (machine failure, for example) occurs, the computer algorithms evaluate the best possible action. Depending upon the desired level of automation, such an action can be reported to the manager or implemented automatically. Algorithms are also applied to areas such as job scheduling and maintenance planning for optimal decision-making. The algorithms are aided by forecasting techniques that use probabilistic prediction from past data to simulation the industry. This also makes it possible to evaluate the costs, downtimes of a given operations plan for the enterprise.

Recent economic and international threats to industries have encouraged companies to increase their performance in any possible way. Many look to rethink their planning systems to quickly react to and correct deviance from established plans, respond to demand, and exchange information promptly throughout the supply chain. Already, innovative companies in some of the world's emerging economies are building "smart" factories: those designed deliver quality products at significantly reduced costs through innovation and production processes radically different from those in place in most industries today. These are not cheap labour-cheap parts solutions; they are entirely new ways of operating industrial plants through the intersection of human innovation and integrated technology.

This project is an effort towards realizing such a future. The aim is to demonstrate the potential of smart manufacturing techniques to industries. It is a system that when plugged into the operations of an industry, offers a complete real-time and remote planning, monitoring and decision-making solution. The availability of such technology is certain to be pivotal in terms of efficiency of operation, making it imperative for any enterprise that wishes to remain competent to adopt this change. Special care is taken to make the project fit for Indian manufacturing industries, and to keep the industry implementation easy and low in cost.

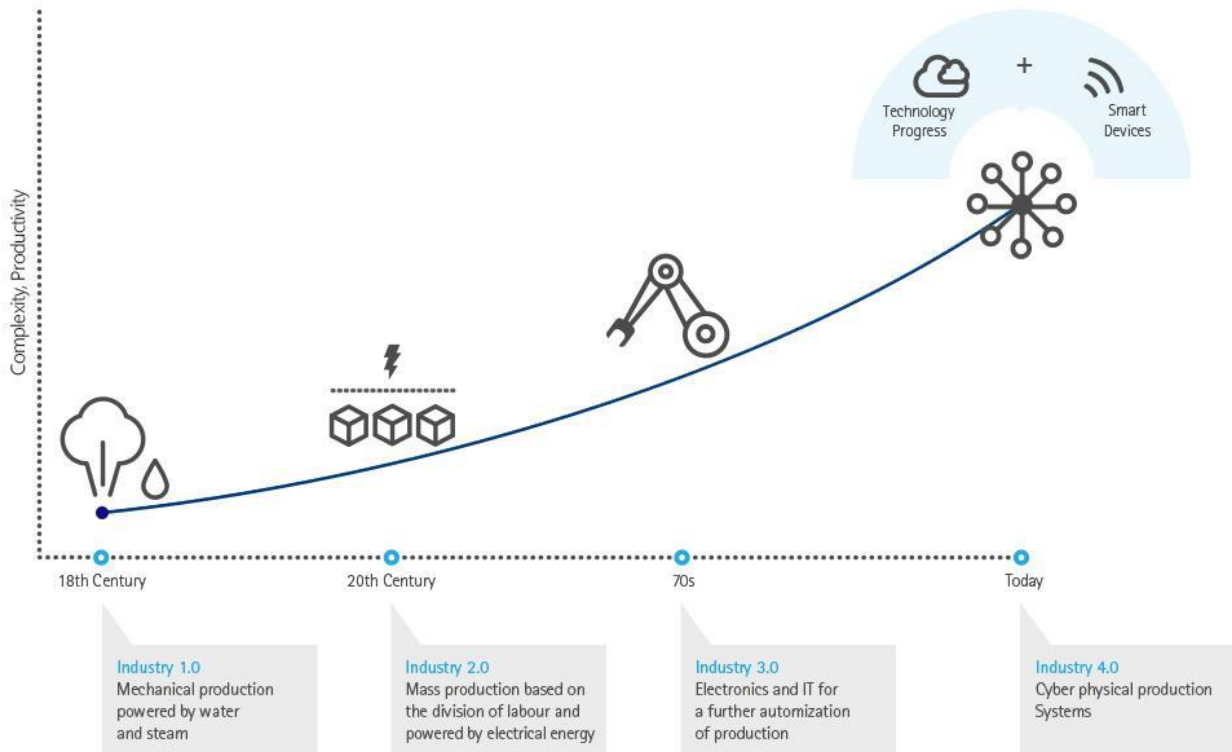


Figure 1. The advent of the 4th Industrial Revolution

Algorithmic and design challenges are plenteous when developing such a system. Several decision variables are combinatorial explosions involving stochastic simulation models. Keeping run-time tractable is a major challenge prevalent in this area. Currently available planning software takes several hours or even days to produce results for relatively simple cases. Moreover, the results produced rely on assumptions that may not always be realistic, or on over-simplified models that fail to reproduce realistic situations. Enterprise models that involve multiple machines and a component level model detailing of each machine get exponentially complicated with increase in model parameters. The project aims to address all of these challenges by adopting a philosophy of decentralization of information as well as decision-making. Concepts of multi-agent systems and social networking are used while modelling the enterprise communication network and decision-making algorithms. There is focus on attributing individual intelligence and human characteristics to enterprise assets. This approach offers several benefits. It allows for consideration of complex cases and detailed models, and it makes the project scalable to increase in problem size or design additions such as new enterprise assets.

The following benefits are expected of the project:

- Make industries lean and more efficient.
- Reduce wastage, downtime, inventory and rejections.
- Managers are connected to the entire enterprise at all times.
- Faster, intelligent and more data-based decision making possible.
- Decision making at elemental level possible.
- Minimize disruptions and prevent monetary loss due to unpredictable events.

Following sectors are the immediate beneficiaries of such a system:

- Small or medium enterprise and large scale manufacturing industry, for example, automobile industry and their sister concerns, would be interested in using such a system for converting their existing shop floor into smart manufacturing system.
- Machine tool manufacturers would be interested in integrating such systems incorporated with their machine tools.
- Software industries would be interested in developing the applications and other software modules, etc. for the proposed system.
- The developed system could also be used to demonstrate Industrial Engineering concepts in a lab environment in educational institutes like IITs and NITs in India. A Virtual Lab can be setup at various institutions to replicate realistic shop floor environment of one industry at each institute, for example. The same can be used remotely by students of other institutes.

The remainder of this section describes the industry scenario being considered, and defines terms that are used in the report.

1.1. Introduction to industry concepts and terms used.

Enterprise: The manufacturing industry that is under consideration. Enterprise is a collective term for all the assets (machines, departments). The term is used interchangeably with *industry*.

Machines: Machines of an enterprise are equipment that can process jobs. They comprise of subassemblies or components. They operate in shifts and there is a schedule of jobs and other actions such as maintenance or precautionary checks that must be performed in a shift. The condition of machines degrades with time and leads to breakdowns due to wear and tear. The number of machines in the enterprise is represented by *'m'* and number of components by *'c'*.

Scheduling Department or Production Department: A department responsible for managing job demands and accordingly creating the job schedule for each machine.

Corrective Maintenance (CM): A machine is said to breakdown when one or more of its constituent components fail. When this happens, CM must be performed on the failed components to restore them to working state.

Preventive Maintenance (PM): To mitigate the likelihood of machine breakdowns due to wear and tear, preventive maintenance or PM is performed on machines. Deciding when to perform PM, picking the machines and choosing the components of the machine to perform PM on is an important decision.

Maintenance Department: The maintenance department is responsible for preparing the schedule for preventive maintenance and managing labour that is required for any CM or PM activity. In general, corrective maintenance is costlier to perform as compared to preventive maintenance. Hence preventive maintenance planning must be done thoughtfully to optimize costs. Too less PM can lead to frequent failures, more downtime, lesser production and high costs due to CM. On the other hand, too much PM leads to high PM costs incurred, and a strain on maintenance labour.

Planning of operations: Several decision variables exist in an enterprise that need to be considered while planning operations. Scheduling of jobs and planning of PM are two such examples.

2. Literature survey and State of the Art

A survey on ‘Smart Manufacturing’ was conducted under the initiative collaborated by ARC Advisory Group, Confederation of Indian Industry (CII) Pune and ISA. 127 companies from various sectors like Process Industry, Machinery and Equipment Manufacturers, Automotive industry, Pharmaceutical, Cosmetics, Chemical, Telecommunications, Aerospace, and Healthcare took this survey. Nearly half of the companies that took the survey are identified as current or potential end users of the outcome of smart manufacturing projects. On analysis of the survey results, some primary business drivers are identified which are fueling the adoption of ‘smartness’ in manufacturing and production, some of them are as listed below:

- Integration of manufacturing processes with other systems like ERP/CRM.
- Reduction of wastages and rejections.
- Better utilization of manufacturing assets, reduced machine or asset downtime.
- Integration of suppliers and other stakeholders into manufacturing processes.
- Integration of all shop floor and other assets into a seamless network.
- Improved and efficient IT driven manufacturing processes.

The survey asked the companies about the drivers that motivated them to work towards adopting Smart Manufacturing. The answers that were given by most of the survey takers were:

- Ensures competitiveness and relevance
- Enhancing bottom lines
- It will boost quality focus
- Want to be early adopters
- Driven by ecosystem and our main clients

Low process and people maturity, lack of technology standardization, budgetary constraints, lack of awareness of intricacies- were found to be the major inhibitors to Smart Manufacturing adoption. It can be concluded from the survey that Indian Manufacturing and production sector has come a long way in the field of smart manufacturing, but has a long way still to go to compete with its international counterparts.

Table 1. Cost due to downtime by sector.

Industry Sector	Revenue Loss/Hour
Energy	*$2,817,846*
Telecommunication	$2,066,245
Manufacturing	*$1,610,654*
Consumer Products	$785,719
Chemicals	$704,101
Transportation	$668,586
Healthcare	$636,030
Electronics	$477,366
Construction	$389,601
Average	**$1,010,536**

Table 1 shows the data of downtime costs by sectors, collected by META Group, Stamford, in their paper "A Comprehensive View of High-Availability Data Center Networking". The Manufacturing and Energy Sectors are areas where the techniques devised in the project find application. The manufacturing sector has assets such as machines whose condition degrades over time, while the energy sector involves critical equipment whose failure leads to extremely high costs. Optimization of the decision variable of preventive maintenance planning has great scope in these areas.

Several companies are offering Smart Manufacturing related products and technologies are commercial solutions. Wipro uses smart devices such as sensors for asset monitoring and wearable devices for improved man-machine interaction. Good effort in collecting, organizing and improving presentation of data is made. It has over 20 products that attempt to embed technology across enterprises. Its Smart Manufacturing segment makes collected data available across the enterprise in real-time and also has the ability to respond to events such as machine malfunction. However these features are limited in scope of applicability, and are mostly aimed at reporting status to a human operator rather than taking action.

The Indian government`s policy on Internet of Things (IoT) includes funding Smart Manufacturing projects. As per the Draft Policy, the objective is to create an Internet of Things (IoT) industry in India of USD 15 billion by 202. Under the domain of Smart Manufacturing, Government desires to use IoT for:

- Planning preventive and in-time maintenance for equipment in manufacturing verticals.
- Process improvement in manufacturing which leads to optimal utilization of resources and
- Monitoring operations and creating warning/alerts for deviation/damages.

Germany, US and UK are leading the research on Smart Manufacturing on the global front, and are working towards realizing the Fourth Industrial Revolution. The German government even coined the term ‘Industrie 4.0’ for this. Germany’s position as an embedded systems leader has given birth to enabling Cyber-Physical System (CPS) technologies which ingeniously marry the digital virtual world with the real world. Industrie 4.0 is one of the 10 future projects identified by the German government as part of its High-Tech Strategy 2020 Action Plan to pursue innovation in the country. The term Industrie 4.0 is defined as 'a collective term for technologies and concepts of value chain organization' which draws together Cyber-Physical Systems, the Internet of Things and the Internet of Services. Within the modular structured Smart Factories of Industry 4.0, cyber-physical systems monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. Over the Internet of Things, cyber-physical systems communicate and cooperate with each other and with humans in real time, and via the Internet of Services, both internal and cross-organizational services are offered and utilized by participants of the value chain.

Rockwell automation has adopted a plan to ‘entirely restructure the facility and supplier network’ using Smart Manufacturing techniques, predicting faster times to market and better enterprise risk management as the result. It divides the plan into three phases: Phase 1- integration of manufacturing data throughout individual plants and across enterprises. Phase 2- Pairing of data with advanced computer simulation and modelling, to create a robust “manufacturing intelligence” that will enable variable-speed, flexible manufacturing, optimal production rates, and faster product customization. Phase 3- Innovations in processes and products that comprise smart manufacturing’s promise — major market disruptions such as a $3,000 automobile or a $300 personal computer.

Though similar software that attempt to aid planning in an industry (such as ERP) exists, it becomes considerably abstruse for medium and smaller firms to adapt to the standards of the system. A high failure rate of such software is reported by companies, for it being too rigid and too difficult to accommodate the specific workflow and business processes, due of the tool's inability to consider the various uncertainties involved in the planning entities and frequent changes in the production plans. Such software actually lacks in the inherent intelligence required for smart manufacturing planning. None of them are able to offer a complete solution that supports at once - distributed organization, generic multi-machine and multi-component systems, integration of machines with software, agility, scalability, and fault tolerance.

Efforts in literature are focussed towards creating models for machines that can simulate machine behaviour, forecast uncertain events and optimize decision variables such as PM planning and job scheduling. Although research on maintenance optimization was established years ago (Dekker 1996), the area of simulation-based optimization is becoming an emerging trend (Garg et al. 2006, Sharma et al. 2011). Rezg et. al. (2005) developed both analytical and simulation models for this problem and found that the analytical model was complex and made unrealistic assumptions, whereas the simulation-based model was more flexible in handling dynamic and stochastic nature of maintenance operations of an enterprise. Hence a stochastic simulation model is used in the project, as it takes into account uncertainties due to failure of machines and variability in time taken for repair, etc. Yang et. al. (2009) described a method for scheduling maintenance operations by assessment and prediction of the level of degradation of machines, and the impact of maintenance operations on production process. This involves evaluating effects of all possible maintenance schedules through discrete-event simulation that utilizes predicted probabilities of machine failures in the manufacturing system. Dhawalikar et. al. (2015) did probabilistic prediction of failure from Weibull analysis of failure data, and used Simulated Annealing as heuristic for reaching a solution.

However, very little work can be found in literature that considers a multi-component machine, which can be vital in improving the probabilistic predictions. The project considers such a model, and is described in detail in the following section. The multi-component machine model allows specifying the number of components, failure and repair distribution characteristics of each component.

As the objective function for such a stochastic simulation model cannot be analytically defined, numerical optimization methods cannot be applied. Hence the problem of optimization in this area usually results into a combinatorial explosion with extremely large search spaces. Several heuristic search methods have been used for problems in this area in literature, mostly commonly – Genetic Algorithms (GA) and Simulated Annealing (SA). While genetic algorithms have been successfully applied to many optimization problems, premature convergence is an inherent characteristic of such classical genetic algorithms (Garg 2009). This project uses an extension of Genetic Algorithm called Memetic Algorithm (MA) to improve search efficiency. Memetic algorithms have been used in several areas such as location optimization of health facility for diabetics (Alegre et. al. 2005), multi-compartment vehicle routing problem (Mendoza et. al. 2010), and maintenance of railway infrastructure (Budai et. al. 2009). The performance of the MAs is found to be better as compared to other techniques like Brute Force Search (BF) and Genetic Algorithms (GA).

Advanced planning and scheduling (APS) systems, that are considered by many as the state-of-the-art of manufacturing and supply-chain planning and scheduling practices are discussed in Stadtler (2004). APS have been developed over time provide better support in complex planning decision making. They are based on the principles of hierarchical planning and make extensive use of mathematical programming and metaheuristics. They work by exploiting operations research, heuristics and constraint programming in order to perform planning and scheduling optimization at the various term levels (long-term, mid-term and short-term) in a supply chain. Long-term decisions include procurement, production, distribution, sales etc.; Mid-term decisions include forecasting, aggregate production and distribution decisions etc.; and Short-term decisions include synchronizing production operations into various plans taking into account technology and capacity constraints while satisfying requirement and as well as plan maintenance. However, different planning tasks are constructed in APS software as segregated individual modules (like Demand planning module, Production planning etc.). Planning in the system is carried out in a time-bucket oriented format, which means that planning is done for the prescribed time-span using accurate forecasted demand, which forms an important input for the decision models. This leads to ‘nervousness’ (dramatic changes in the plans) if there are further adjustments in the forecasted information while the devised plan is being executed. Instead, an event scheduling scheme will be more appropriate where the given plan is updated whenever new information comes in.

Forget et. al. (2008) attempt to develop a new conceptual agent model, with the objective to describe the characteristics needed to enhance agility and synchronization of agents developed in planning systems like the FOR@C experimental platform (Frayret, 2008). They support the hypothesis that agility and synchronization can be improved by implementing adapted behaviours (or strategies) depending on the execution conditions. To deploy agents with different behaviours, agents must possess the ability to make choices and the capability to evaluate these choices following specific criteria or goals. These abilities to make decisions are addressed in the form of competency levels. Integrating agent technology and tools, the model is composed of three distinct layers, describing different competencies required for supply chain planning: Technical Competencies, Decision Competencies and Social Competencies.

The bottom layer of the agent model consists of the Technical competency layer. This decision layer includes all tools, tasks, and existing Task Flows (TFs), such as algorithms, conversation protocols, negotiation protocols, and queries. The Social Competency layer serves the purpose of integrating the welfare of partners through collective goals. The agent is now aware of the impacts of its decisions on other agents and on the whole supply chain. While choosing actions to correct deviations from plan, the agent possess the ability to capture the entire potential of the network and be able to minimize impact on others. Strengthening the above stated competency layers, the Learning Competency, embedded in each layer, gives the agent potential to increase its knowledge. A specific action or sequence of actions that demonstrated positive results in a situation could be learned and remembered for the next occurrence.

Multi-behaviour agent model is an evolution of the described concept. The model presents three basic behaviours to react to a new state in a planning context: Reaction, Anticipation, and Negotiation. Under Reaction behaviour, agent knows a certain number of TFs and can use one of them to respond to a disturbance. Different optimization algorithms and objective functions can be used, depending on the situation and on the available time. The agent use only what is the best for him. No knowledge about partners is used and there is no way to check if the proposed plan will satisfy the partner. The Anticipation behaviour is a planning strategy using a partner model in addition to its local model. Basically, it is about integrating information about partners into its optimization model. Depending on the situation, emphasis can be placed on local or collective goals. The Negotiation behaviour describes TFs sending proposals to partners, in the form of alternative plans. When the agent is not able to respond to partner's needs, it can offer changes in delivery dates or alternative products. Following this, an iterative exchange of proposals is started, where both agents try to find a compromise.

The main advantage is the possibility of adjusting the behaviour according to external factors. For example, when a client sends a demand plan and requests an acceptance or a refusal in a short time frame, the agent is able to use its fastest response, which is one of the reaction TFs. In contrast, if a large amount of time is available, the agent would take time to send new demand plans to suppliers. Another advantage is the possibility for the agent to use collective goals in addition to local goals. Anticipation TFs use inputs from a partner's model in order to integrate both local and collective goals. Depending on the relative importance of these goals, a balanced solution can be reached. Although this description of advantages seems promising, it is still based on an untested agent model. Moreover, high level of computation will be involved in real life application of the presented system. Performing it in a single computational system seems less feasible. Instead a distributed computation can be performed with more ease.

In Yang et. al. (2007), a stochastic model based on historic data is used to simulate a multi-machine industry for maintenance scheduling. The part of the model that is used to predict uncertainties is similar to the one employed by the project. A Genetic Algorithm based optimization procedure is used to search for the most cost-effective maintenance schedule, considering both production gains and maintenance expenses. In all cases that were studied in the paper, application of the newly proposed maintenance scheduling tool resulted in a noticeable increase in the cost-benefits, which indicates that the use of predictive information about equipment performance through the maintenance scheduling method results in significant gains. But the machine model does not allow description of detail to the component level. This is handled in Tambe et. al. (2013), where a stochastic model of multi-component machine is used for simulation. However, both these works still employ centralized approaches that are limited in several ways as compared to distributed or decentralized approaches.

According to the Intelligent Software Agents group at Carnegie Mellon University, a Multi-Agent System (MAS) has the following advantages over centralized approaches:

- An MAS distributes computational resources and capabilities across a network of interconnected agents. Whereas a centralized system may be plagued by resource limitations, performance bottlenecks, or critical failures, an MAS is decentralized and thus does not suffer from the "single point of failure" problem associated with centralized systems.
- An MAS allows for the interconnection and interoperation of multiple existing legacy systems. By building an agent wrapper around such systems, they can be inporporated into an agent society.
- An MAS models problems in terms of autonomous interacting component-agents, which is proving to be a more natural way of representing task allocation, team planning, user preferences, open environments, and so on.
- An MAS efficiently retrieves, filters, and globally coordinates information from sources that are spatially distributed.
- An MAS provides solutions in situations where expertise is spatially and temporally distributed.
- An MAS enhances overall system performance, specifically along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse.

2.1. Observations from survey:

- Several industries global and Indian industries are making an effort to overhaul the way the industry works by using Smart Manufacturing technologies. The reason for this is the improvement in efficiency, better process quality, reduced losses, increased competitiveness that such technologies promise.
- The true potential of distributed planning is not unlocked by any of the current works. Most planning tools employ centralized methods that operate on a single computer system (sharing the same computing resources) resulting in long evaluation times, and impose severe limitations on the level of detail of the model, and hence the efficiency and usability of the devised operations plan. Distributed simulation that penetrates into individual (but significant) constituents of the planning process is yet missing.

- Current state of art lacks the robustness and fails to successfully take variability into account. Some systems use mathematical or deterministic models which ignore context that addresses variability of the process parameters (such as processing time of an activity) or the chance of occurrence of unexpected events (eg. failure of a machine). Developing detailed stochastic models for addresses this limitation.
- Current state of the art is not flexible in updating the devised operation plan based on real-time events such as failure or changes in operational parameters. A strategy is necessary where the given plan is updated whenever new information comes in. The ability of the project's simulation model to interface physical entities such as machines with the simulation models will serve a pivotal role.
- Genetic Algorithms are the most popular heuristic methods used for problems that involve stochastic models and picking a solution from a large domain. They are easy to implement and produce good results.

2.2 Problems and scope for improvement:

- Manual recording and handling of various data (like the number of jobs produced, details of maintenance activities performed, percentage of non-conforming jobs, available level of inventory, etc.) through log books and databases.
- Tedious and time consuming processes for production and maintenance planning, which involves brainstorming by individual personnel from multiple departments.
- Physical identification of critical spares or consumables based on the requirement rate and procurement time.
- Experience based labour allocation plan, which is generally prepared by supervisor/manager handling the machine, line or the production cell, that is , mostly, not the most optimal utilization strategy of the available manpower.
- Manual indication of a machine breakdown by the machine operator to the maintenance department, which incurs an increase in the downtime of the machine.
- Higher wastage in terms of down time, rejection, over-maintenance, etc.

3. Solution Approach

This section describes the philosophy used while designing solutions. In the sections to follow, the use of these approaches has been stated in detail.

3.1. Cyber-Physical Systems or Cyber Twins

Cyber–Physical systems (CPSs) are integrations of computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. Assets in a manufacturing enterprise such as machines, and departments are modelled as CPS or *‘Cyber Twins’*.

The Cyber Twin or virtual entity is interfaced with the physical asset as an embedded system, allowing the virtual entity to receive the real-time state of the physical asset. This confers the advantage of being able to monitor, communicate events and take action briskly.

However these advantages are possible by leveraging sensor technology and embedded systems, which can be costly, difficult, or at times completely infeasible. Hence provision is made for an interface to be provided for human operators. The operators can make use of this to update the state of the asset. The interface is a smartphone application or mobile website that can be easily implemented without needing any additional resources in the enterprise. This makes implementation of the project in an industry easy and universal. More importantly, industries can use such an interface to test the project and its impact on cost and operations. Based on this, the industry can decide if it should invest in sophisticated sensing technology.

Specifically, the Cyber Twin can be described as follows:

1. It is an exact virtual replica of a physical asset of the enterprise.
2. It emulates the states that the physical asset is in, and has cost models that can emulate the costs incurred as the physical asset progresses through these states.
3. It interfaces with the physical asset using embedded sensors or human operator input to track the state of the
4. It is social. It has the capability to communicate with other Cyber Twins in the enterprise and learn and share relevant information.
5. It has models that can be used for simulating the operation of the physical asset.
6. It is intelligent. Using all the capabilities 1-5, it can make autonomous decisions and perform planning tasks.

The purpose of such a CPS model is to be able to gather data for assets and use such gathered data to stochastically forecast uncertainties such as machine failure, time to repair, availability of resources, fluctuations in demand etc. The level of detail of the model (how closely the actual entity is modelled) determines how accurate the predictions made by the model are, and how robust the model can be in handling change in operational parameters. However, the more detailed that a model is made, the greater is the time taken for the planning process. Hence employing a decentralized approach to model an enterprise is important for run-time tractability and scalability. As opposed to a top-down system where a central authority conveys plans to assets, each asset is made intelligent with the capability of being able to take its own decision. In such a bottom-up approach, freedom of individual action selection is constrained by some obligatory forces that become operative in the individual decision making process even without a controlling authority. Being interdependent, the assets must also be able to efficiently communicate with other relevant assets about status, events and decisions.

Hence the Cyber Twin is a fundamental part of the system as it offers functionalities necessary for smart industry management, at modular level. The synergy of such Cyber Twins form the virtual enterprise model.

3.2 Multi-agent system

A multi-agent system (MAS) is a computerized system composed of multiple interacting intelligent agents within an environment. Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent to solve. Agent-based systems technology promises a new paradigm for conceptualizing, designing, and implementing solutions. This promise is particularly attractive for creating software that operates in environments that are distributed and complex, such as the enterprise. As technology matures and addresses increasingly complex applications, the need for systems that consist of multiple agents that communicate in a peer-to-peer fashion is becoming apparent. Central to the design and effective operation of such multiagent systems (MASs) are a core set of issues and research questions that have been studied over the years by the distributed AI community.

The characteristics of MASs are that:

1. Each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint.
2. There is no system global control.
3. Data are decentralized.
4. Computation is asynchronous.

The Cyber Twins serve as intelligent autonomous agents in the MAS model of the enterprise.

Figure 2. provides the overview of the project. The problem can be divided into the following sub-problems:

1. Modelling of intelligent and social cyber-twins for each enterprise asset.
2. Creating a network of such cyber twins for flow of information.
3. Creating intelligent algorithms and communication methods for planning and decision-making.
4. Developing a software and webapp for implementation of the system in industry.

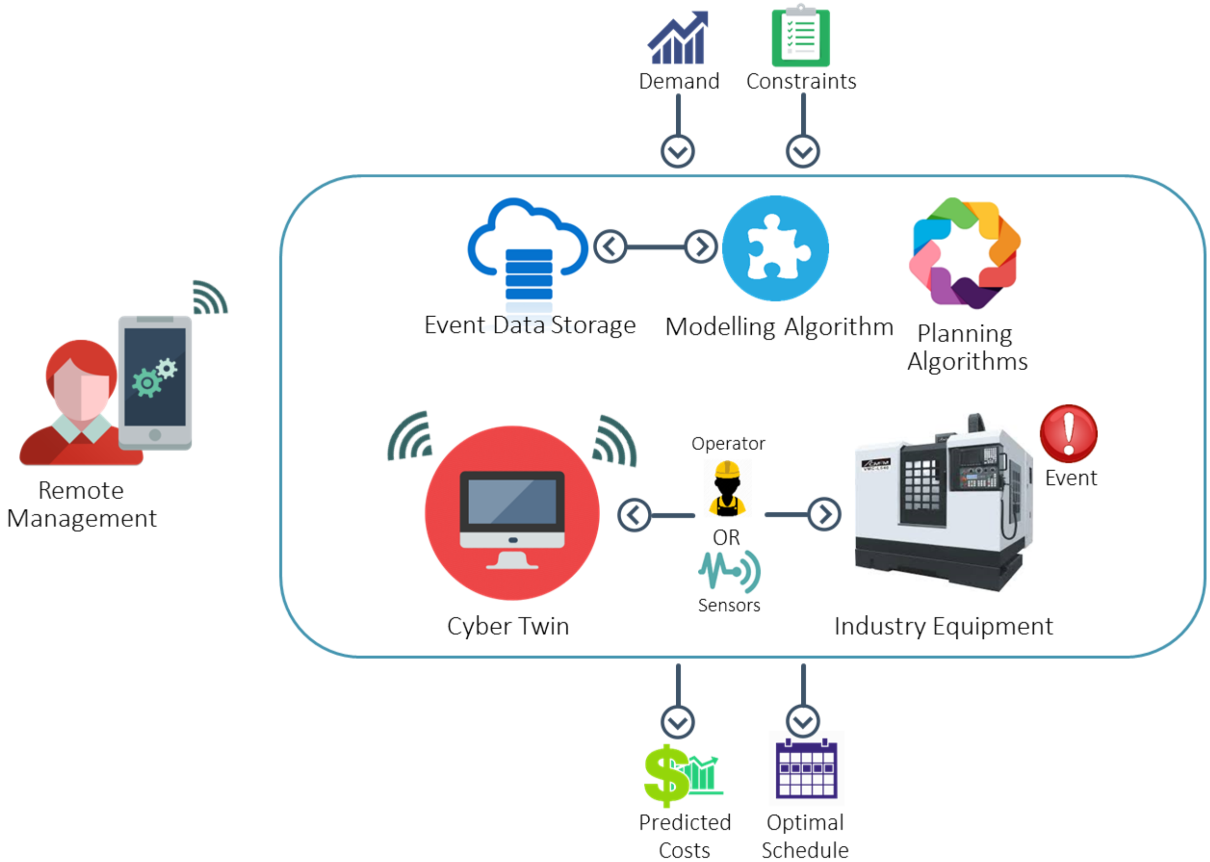


Figure 2. Overview of project

4. Cyber Twins

4.1. Machine Cyber Twin

Consider a machine consisting of multiple sub-assemblies, with each sub-assembly comprising of multiple components. Let the total number of components in the machine be represented by *'n'*. Initially, a component may not be new and could have an accumulated age. A machine is said to breakdown when one or more of its constituent components fail. It is assumed that the components have only one failure mode, which brings the machine to breakdown state instantly and can be detected immediately. It is also assumed that the failure of a component is independent of other components.

Table 2. Model Parameters for Machine Comprising Components C1, C2, C3 and C4.

		Corrective Maintenance (CM)									Preventive Maintenance (PM)							
		Reliability		Maintainability		Supportability			Fixed Costs		Maintainability		Supportability			Fixed Costs		
COMP	AGE (hr.)	Time to Failure (hr.)		Time to Repair (hr.)		Waiting time for resources (hr.)		RF	Spare Parts (Rs.)	Other (Rs.)	Time to Repair (hr.)		Waiting time for resources (hr.)		RF	Spare Parts (Rs.)	Other (Rs.)
	X	η	β	μ	σ	μ	σ	α			μ	σ	μ	σ	α		
C1	9000	4000	2.3	4	1	1	0.5	1	40000	0	4	1	2	0	0.1	0	1000
C2	9000	4000	2.3	6	1	3	0.5	1	36000	0	6	1	1	0	0.5	300	9000
C3	9000	4000	2.3	5	1	2	0.5	1	38000	0	5	1	2	0	0.4	100	9500
C4	1000	4000	2.3	4	1	1	0.5	1	40000	0	4	1	1	0	0.3	0	10000

The data of failure times and repair times of a component are collected and fitted into a distribution. The failure time is observed to follow a two-parameter Weibull distribution (shown in Figure 3) with a given shape parameter (*β)* and scale parameter (*η)*.

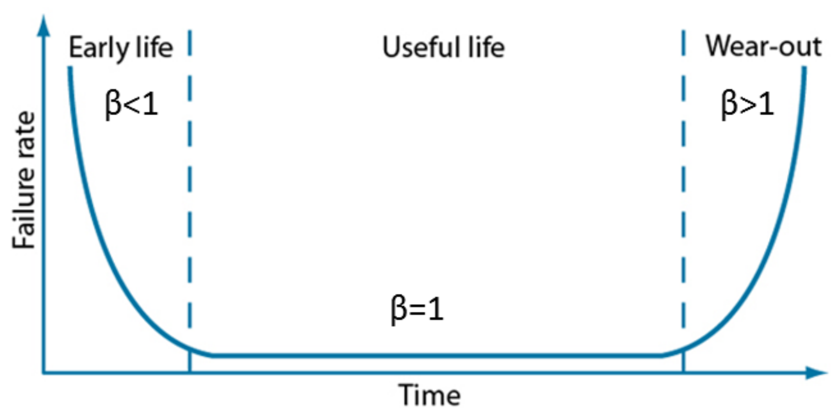


Figure 3. CM TTF for a component follows the Weibull Distribution

Upon failure, Corrective Maintenance (CM) activity is performed on the machine. The total time for a CM activity for a component is given by the sum of actual time taken to repair (maintainability) and the time required to arrange for resources to perform repair (supportability), for that component. The uncertainties related to repair actions make it difficult to precisely predict the time required for CM activity. Hence, the time to repair (TTR) for CM is considered to be normally distributed with the mean (μ) and standard deviation (σ). μ and σ for a component can be easily obtained from its past data of TTRs.

Table 2 describes the various model parameters for a machine comprising 4 components C1, C2, C3, C4. The parameters η, β, μ and σ for a component are obtained from one of the following sources:

- Fitting historic data into probability distributions.
- Provided by manufacturer of the component.
- Obtained from standard reliability data.
- From the parameters of similar components.

After a maintenance activity has been performed, age of the component is reduced by a factor known as Restoration Factor (RF) for that component. RF is used as a measure to model the degree of repair of a maintenance activity. A restoration factor of 1 signifies maximal or perfect repair. It means that the age of the machine component is restored by 100% to zero – 'as good as new condition' - after the maintenance activity. On the contrary, a restoration factor of 0 signifies minimal repair. This means that there is no effect of the maintenance activity on the machine component's age and thus the component stays in an 'as bad as old' condition. A restoration factor between 0 and 1 signify an imperfect repair with the value of RF signifying the extent to which the age will be repaired after maintenance activity. For example, for a component of age 1000 hours, a maintenance activity with the restoration factor of 0.8, will reduce the age to 200 hours. The RF of a CM activity for a component is taken as 1, which is the case of replacement of failed component with a new one. Moreover, every component has specific labour requirements in terms of quantity and skill of the labour. Skilled, semi-skilled and unskilled labour are available for maintenance action, each incurring different costs, to carry out a maintenance activity. It is assumed that labour is always available for maintenance when it is required.

Preventive maintenance (PM) is a scheduled maintenance activity done as a proactive measure to mitigate machine failure and downtime losses. The total TTR for PM of a component, follows a normal distribution, similar to TTR for CM described above. As a PM activity represents imperfect repair, the RF for a PM activity is taken between 0 and 1. Further, if a maintenance activity involves more than one component, then it is performed as a series of maintenance activities (done one after the other) for each involved component. The cost incurred to carry out a maintenance activity consists of fixed costs (cost of spare parts, and other fixed costs) and cost of maintenance labour.

The machine receives a job schedule for each shift, which is the description of jobs that it must complete in that shift. Each job has a due date and a penalty cost per hour, for every hour of delay after the due date. A delay in the job can be caused by machine breakdown or preventive maintenance activity. A PM activity can be scheduled after a job has ended and before the next job has begun, i.e. in between two jobs, but cannot interrupt a job. The instances in a schedule when a PM activity can be scheduled are referred to as *PM opportunities*, and the number of PM opportunities in a schedule is denoted by *'m'*. *m* is equal to the number of jobs in the schedule. PM activity must be incorporated into the schedule such that the total cost incurred in the shift is minimum. This involves answering two questions (i.e. decision variables) - when (at which opportunities) to perform PM and on what components to perform PM for an opportunity?

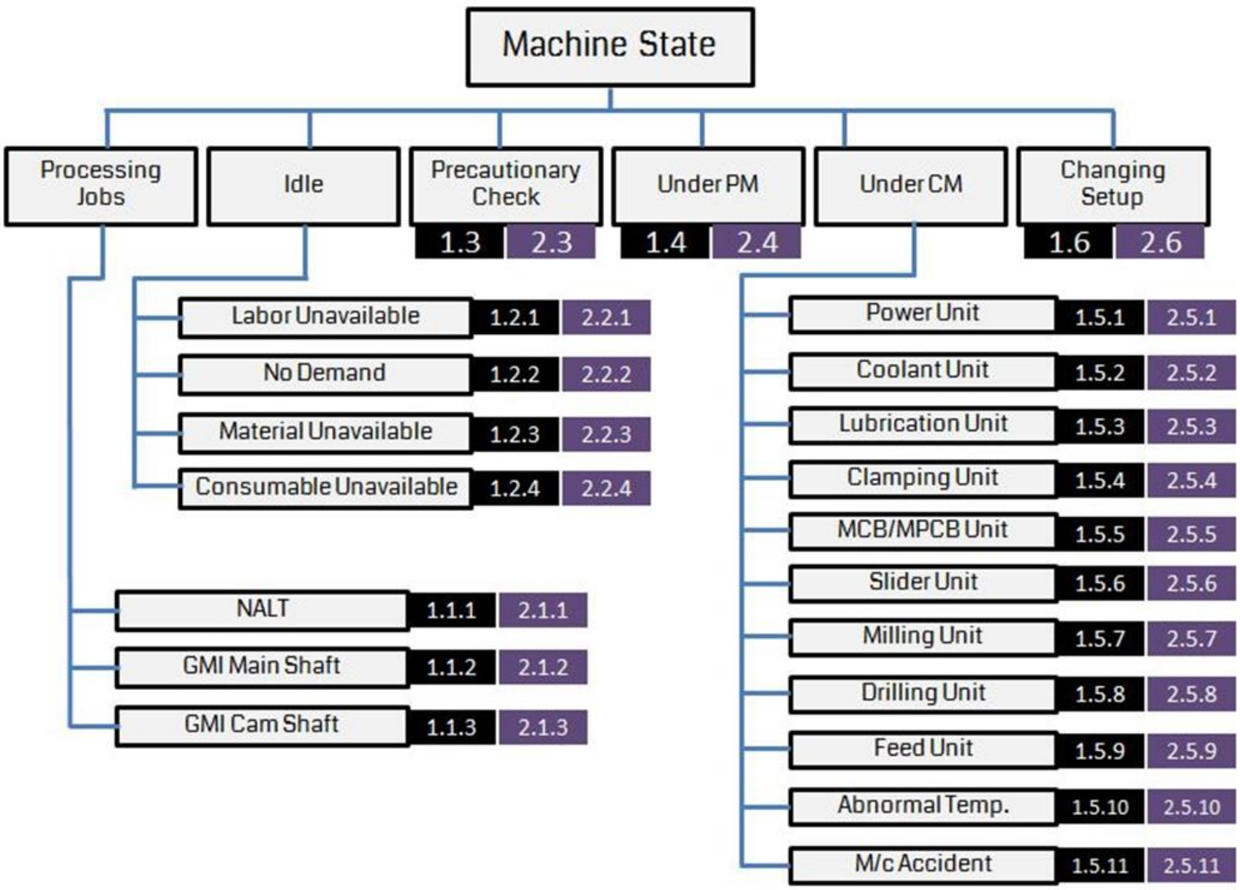


Figure 4. Modelling of Cyber Twin for HMT Facing and Centering Machine at AVTEC Ltd.

Figure 5. HMT Facing and Centering Machine at AVTEC Ltd.

Figure 4 depicts the Cyber Twin model of an HMT Facing and Centering Machine (shown in Figure 5) at AVTEC Ltd. The machine processes Cam Shafts. Each possible state that the machine can be in, is documented, and attributed to the Cyber Twin. The components of the machine or modes of failure listed under the 'Under CM' block.

4.2. Production or Scheduling Department Cyber Twin:

The production or scheduling department is responsible for collecting job demand and conveying job schedule to the machines. Once the demand is generated for a shift, the scheduling department schedules the jobs to the machines using the Longest Processing Time First (LPT) rule. LTP rule is a common heuristic used in parallel machine scheduling problems. Table 3. depicts the description of jobs along with the parameters for each job, in Scheduling Department Cyber Twin.

Table 3. Description of Jobs and Job Parameters.

Job ID	Processing Time(hr.)	Due Date (hr.)	Penalty Cost (Rs./hr.)
J1	206	8	500
J2	195	8	500
J3	193	8	500

4.3 Maintenance Department Cyber Twin:

The maintenance department Cyber Twin is responsible for planning preventive maintenance and managing labour for any maintenance task. The model parameters are described below in Table 4.

Table 4. Description of maintenance labour considered in model.

	SKILLED	SEMI-SKILLED	UNSKILLED
Cost (Rs./Hr.)	800	500	300
Available	1	2	4

The entire process of production in the manufacturing unit starts with the scheduling department generating a schedule for the shift immediately before the shift begins. This scheduling is done for individual machines using the LPT Rule and conveyed to respective machines. Then the machines identify different opportunities for performing PM and compute its 'interest' in undergoing PM at every opportunity. Furthermore, it even decides the combination of components which needs to be maintained together. These decisions are taken by a comparison between overall costs incurred in production for different maintenance scenarios for that shift. Moreover, every 'perform-PM-decision' has a certain Intensity Factor (IF), which accounts for the level of 'interest' of the machine in performing the action. Ultimately, the machine level PM schedule with Intensity Factors, the average time required for performing the PM and the total number of labourers required to perform the PM are sent to the maintenance department. Then, it is the job of the maintenance departments to come up with a unified schedule for performing preventive maintenance for all the machines. To do this the maintenance department machine level arranges all the machine level PM schedules in decreasing order of IFs and then the total labour required and next the average PM time for that schedule. After ordering the schedules, the maintenance department assigns one machine level PM schedule to each machine only if it is able to meet the labour requirements. Once the overall schedule has been made and sent back to the machines, the machines start executing their respective schedule and request the maintenance department for labourers whenever required. In case a machine breaks down, the maintenance departments dispatched labourers to the machine as soon as the required labourers are available.

4.4. Cost Models and Simulation

The cost models for all the various costs involved are represented by equations (1)-(3). The models have been adopted from Lad and Kulkarni (2013). The total cost for a shift is given by the sum of delay costs of all jobs, costs incurred due to all PM activities and costs incurred due to all CM activities upon simulating the schedule, is given by equation (3). *'i'* denotes the i^{th} component, *'j'* denotes the j^{th} PM opportunity and C_{ij} denotes if PM is performed on the i^{th} component in the j^{th} opportunity, it is either 1 (yes) or 0 (no).

$$\begin{aligned} &Maintenance\ cost\ for\ a\ component \\ &\qquad = (Total\ Fixed\ Cost) + (Time\ to\ repair) \\ &\qquad * (Total\ Labour\ Cost/\ Hour) \end{aligned} \qquad \ldots(1)$$

$$Penalty\ cost\ for\ a\ job = Penalty\ cost\ per\ hour * Delay\ in\ hours \qquad \ldots(2)$$

$$\begin{aligned} &Total\ Cost\ Incurred\ in\ a\ shift \\ &\qquad = Total\ CM\ Cost \\ &\qquad + \sum_{i=1}^{n}\left(\sum_{j=1}^{m} C_{ij} * PM\ Cost\ for\ i^{th} component\right. \\ &\qquad \left. + CM\ Cost\ for\ i^{th} component\right) + Total\ Penalty\ Cost\ for\ all\ jobs \end{aligned} \qquad \ldots(3)$$

$$where\ C_{ij} = \begin{cases} 1\ if\ PM\ is\ performed\ on\ i^{th} component\ at\ j^{th} opportunity \\ 0\ if\ no\ PM\ is\ performed\ on\ i^{th} component\ at\ j^{th} opportunity \end{cases}$$

To perform cost based optimization, a method is needed to evaluate the cost of operation of the machine for any given shift schedule. Monte Carlo Simulation is used for this purpose, and the total cost incurred for a schedule is obtained by averaging the costs obtained by simulating that schedule multiple times. The number of times a schedule is simulated to obtain the average cost incurred is referred to as simulation count, denoted by *'k'*.

To test and validate the devised plan, a prototype is designed which simulates the behaviour of the agents mentioned above, namely the multi-component machine, the scheduling department and the maintenance department. The entire simulation framework has been written in Java in a modular fashion. The different agents communicate with each other via TCP sockets to ensure reliability. The machine simulator mimics the behaviour of a real machine. Time of failure of and time to repair for a machine follow probability distributions that are characterized by past data of time of failure and time to repair.

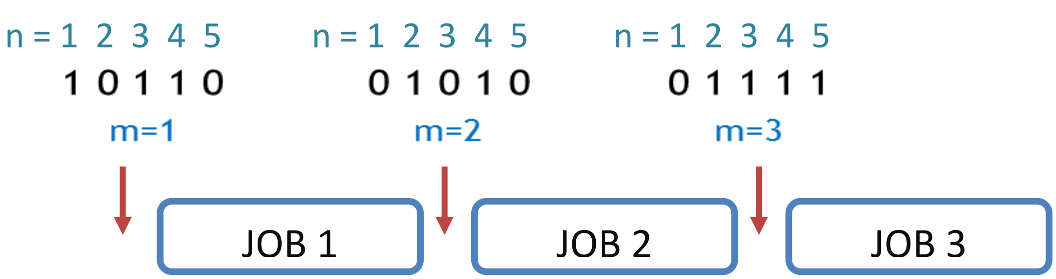
The scheduling department generates demand randomly and schedules the jobs to the machines. The whole framework runs on different computer systems in a distributed manner (each agent on a different computer) and displays the execution and planning phases of the agents in real time on a central logging system. Time synchronization between different agents is achieved by periodically polling a central system. It possible to interface an actual machine with the machine simulator. In this case, instead of using probability distributions to simulate uncertain events, information about the machine state and such events is picked up directly from the sensors of the machine.

5. PM planning for a multi-component machine

Out of 2^{m*n} possible solutions in the solution domain, the solution S_i must be found such that it has the least average cost. Hence the objective function can be defined as:

$$\min_{\forall i \in [1, 2^{m*n}]} S_i$$

Where S_i is the average of total costs incurred when the i^{th} possible PM incorporated schedule is simulated *k* times. The solution (PM incorporated schedule) is encoded in the form of a binary string of length *m*n*, as at every PM opportunity there is an option of either performing or not performing a PM for each component. For example, a possible solution for a machine with *n*=5 components and *m*=3 opportunities (3 jobs) can be represented by a binary string as:



Each group of bits represents a PM opportunity in the schedule, with the leftmost group being the first opportunity. The value at the i^{th} place, in each PM opportunity tells whether to perform PM (C_{ij}=1) or not to perform PM (C_{ij}=0) for the i^{th} component at that opportunity. In this string, PM activity has to be performed for component no. 2, 3 and 5 at the third opportunity; for component no. 2 and 4 at the second opportunity and for component no 1, 2, 3 and 4 at the first opportunity. The approaches used to solve the problem are presented below.

5.1. Brute Force Algorithm (BF)

Every solution in the solution domain is evaluated to find its average total cost by simulating it k times. The solution with the least cost is the most optimal solution for that shift, and is the solution to the problem. Hence the solutions produced by BF can be used as a benchmark for the optimality of solutions produced by MA.

5.2. Memetic Algorithm (MA)

The flow of the Memetic Algorithm is described in the diagram below and the various operators involved have been explained.

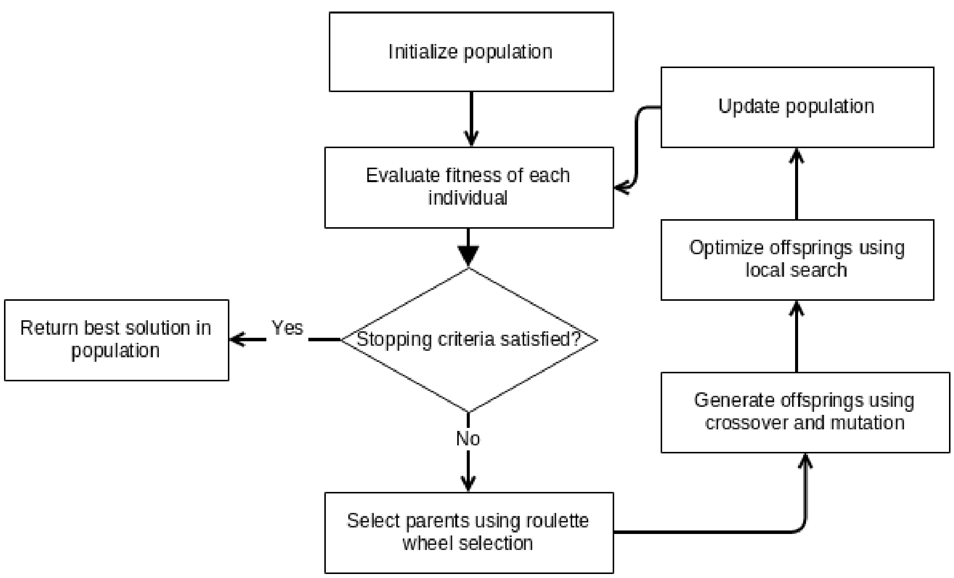


Figure 6. Flow of Memetic Algorithm

Chromosome: The component combinations undergoing PM across opportunities for a schedule in a shift are represented by a binary string as described above. This binary string forms the *chromosome* for the MA.

Initialize population: A fixed number (population size) of chromosomes are randomly taken from the solution domain as the *initial population*. Population size is taken as (2 X *m* X *n*).

Fitness evaluation: Evaluation of the fitness function of the chromosomes in the population is done. For the problem at hand, the fitness function is the average cost of simulating the schedule represented by the chromosome for the shift, *k* times.

Selection of parents: Roulette wheel selection is used to select individuals for reproduction. Number of individuals selected is (m X n) i.e. half of the population size.

Crossover: Pairs from the pool of selected parents are picked randomly and offsprings are generated using Single Point Crossover. For each pair, a crossover point is randomly selected and two offsprings are generated by swapping the chromosome data beyond the crossover point.

Mutation: Mutation is carried out on the off-springs using the bit flip mutation operator. Two bits in the chromosome are randomly chosen and flipped. For this experiment, the mutation probability is 0.4.

Local search optimization: Local search optimization is performed on each individual of the population. A neighbour is defined as any solution that can be reached from the current solution by flipping any two bits in its chromosome. Stochastic hill climbing approach is used to iterate through the neighbours of the current solution, and a neighbour is chosen to replace the current solution only if it results in an improvement. The heuristic function used to evaluate a solution for local search optimization is described in detail in section 4.2.1.

Termination criterion: The search for optimal solution is terminated after 100 generations or if the fittest solution (one with the least cost) is the same for 25 consecutive generations. After meeting the termination criterion the algorithm returns the fittest solution.

5.2.1. Local Search Optimization and Heuristic Function

In local search, an attempt to improve the fitness of the population in each generation is made. The value of heuristic function of a chromosome is compared with the value of heuristic function of its neighbouring chromosomes. If a neighbour has a better (higher) heuristic function value, then the first chromosome is replaced with the neighbour.

Let *i* be the component number, *j* be the PM opportunity. P_i is the probability of failure for the i^{th} component in a shift. Pi is estimated by computing 50 times the time to failure (TTF) for a component (that follows Weibull distribution characterized by η, β and age) and checking how many times the TTF lies in between shift duration. C_{ij} is a bit representing the decision of performing PM, if C_{ij}=1 then PM is performed and C_{ij}=0 then PM is not performed for the i^{th} component in the j^{th} opportunity. The heuristic function is modelled in such a way that a favourable decision regarding performing PM for a component for an opportunity is rewarded (the value of the function is increased by 1), while an unfavourable decision is penalized (the value of the heuristic function is decreased by 1).The heuristic function *H( )*, for a chromosome is given by:

Where,

$$V_{ij} = \begin{cases} \mathbf{1} & \text{,if } P_i > 0.5 \text{ and } C_{ij}=1 \text{ (performing PM on } i \text{ when probability of failure is 'high')} \\ \mathbf{-1} & \text{,if } P_i > 0.5 \text{ and } C_{ij}=0 \text{ (not performing PM on } i \text{ though probability of failure is 'high')} \\ \mathbf{1} & \text{,if } P_i < 0.5 \text{ and } C_{ij}=0 \text{ (not performing PM on } i \text{ as probability of failure is 'low')} \\ \mathbf{-1} & \text{,if } P_i < 0.5 \text{ and } C_{ij}=1 \text{ (performing PM on } i \text{ though probability of failure is 'low')} \end{cases}$$

The threshold value for considering the probability of failure as 'high' is taken as 0.5 here. This can be set depending on the risk of machine failure one wishes to incorporate in the planning.

Consider the chromosome, 101 100, as an example. The machine has three components and the schedule has two opportunities in a shift. Let P_1=0.8, P_2=0.4, P_3=0.6. Then the value of heuristic function is [1 + 1 + 1] + [1 + 1 + (-1)] = 4. A higher value of *H()*is indicative of a more favourable schedule.

5.3. Genetic Algorithm (GA)

To highlight the necessity of the Memetic Algorithm, the local search is turned off to get an equivalent genetic algorithm.

5.4 Experiments and Results

The following setup was used for execution and testing of the three algorithms:

Intel(R) Core i7-4790 CPU @ 3.60GHz

4GB Memory

Windows 7 Professional 64-bit

JDK 8 for compilation and JRE 8 for execution (Java 8).

Table 5. Results for comparison of MA and BF for small problem size.

		Brute Force Algorithm (BF)			**Memetic Algorithm (MA)**		
m	***n***	**Cost**	**Time (sec)**	**Solution**	**Cost**	**Time (sec)**	**Solution**
3	4	36041	59.84	0000 0001 0110	36505.1	7.17	0000 0000 0110
3	5	52029.8	863.66	00000 00000 00110	52818.4	10.49	00000 00000 00110
4	4	37420	1336.51	0001 0000 0000 0110	35633.4	14.75	0000 0000 0000 0110

Table 5 presents the results of BF and MA, both being applied to the same schedule for a shift. Since BF searches the entire solution domain, the solution produced by it can be considered to be the most optimal solution. Hence for a given shift and schedule, this solution can serve as a benchmark for evaluating the optimality of the solution produced by MA. As seen in the table, the solutions produced by MA are of nearly same cost as compared to the solutions produced by BF, if not the same solution.

In the third test case, cost of the solution produced by MA is less than that of the solution produced by BF. This may appear misleading, since the BF is expected to produce the most optimal solution. However, as the cost of a solution is calculated by a stochastic model, it is possible for the same solution to have a different cost each time it is calculated. Hence for a solution, the MA may calculate a lower cost as compared to BF.

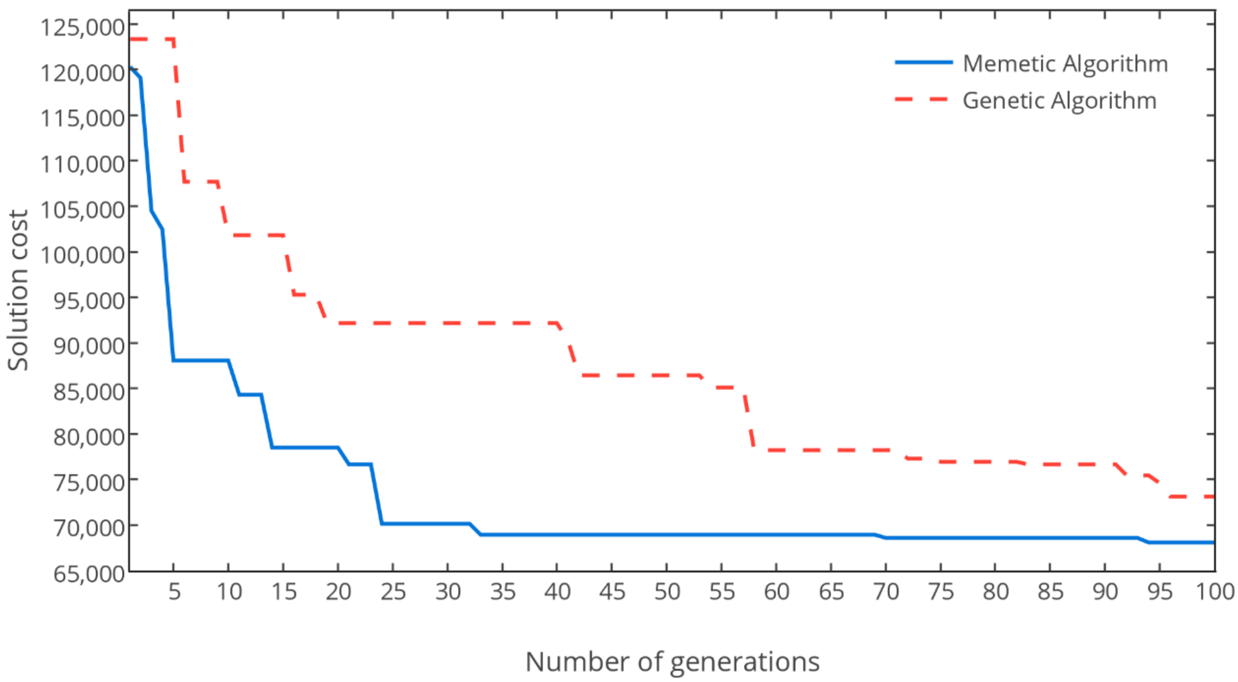


Figure 7. Comparison of performance of MA and GA

For comparing the performance of the MA with that of GA, both algorithms are run for 100 generations (instead of the using termination criteria mentioned in Section 4.2), with the same initial population. The cost of the best solution obtained in each generation is plotted against the generation number in Figure 7. It can be observed from the graph that the MA progresses faster than GA, and is able to reach a better solution in lesser number of generations. This can be attributed to the fact that GA gets stuck at local minima (at generations 18, 41 and 57). On the other hand, the MA produces fitter offsprings due to the local optimization and is able to escape such local minima.

6. PM planning for an Enterprise

Consider an enterprise consisting of a scheduling department, a maintenance department and machines. The machines are arranged in parallel and are identical. Each machine comprises of multiple components. The failure of these components is responsible for bringing the machine down. Let '*m*' denote the number of machines and '*c*' denote the number of such components inside a machine. The initial age of each constituent component is different at the start of the planning horizon considering that each component is not equally involved in an operation during one cycle time.

Let the Maintenance department consist of three categories of labour: High Skilled Labour, Medium Skilled Labour and Low Skilled Labour. Every component inside a machine requires a specific number of labour from each category for carrying out maintenance tasks. Upon the event of failure, corrective maintenance is performed on the machines based on availability of the stipulated labour. If the labour requirements are met, the time to repair for a component is assumed to follow a normal distribution with mean, μ_{CM} and a standard deviation, σ_{CM}. The corrective maintenance action results in a minimal repair of the component i.e. the age of the component after maintenance is same as that before maintenance. Such a condition can be represented by Restoration Factor, $RF_{CM} = 0$. Until the labour requirements for the corrective maintenance activity is met, the system results in waiting time for maintenance.

We consider the system to be of a manufacturing facility which produces parts with long processing times such as aviation components, turbine parts etc. Let each machine be equally capable of processing the job with the processing time be given by a uniform distribution between 460 hours and 500 hours irrespective of the machine. Due to the nature of the jobs being considered, long processing times are attributed to the significant times involved in loading of the job, alignment and tool setup and unloading of the job apart from operation time. Let the demand for a period of 2 months be '*D'* quantity of jobs.

Let us consider a planning horizon of 2 months for scheduling preventive maintenance activities. Since the type of jobs being considered involve long setup times, it is preferable to identify occasions between two jobs, i.e. after the unloading of the previous job and before loading of the next job, as feasible opportunities for performing PM. This would prevent extended downtime in the form of long unloading times for the ongoing jobs before the beginning of the PM activity and loading times of the jobs after its completion if a loaded job is interrupted. Let us identify these periods as 'PM

Opportunities', denoted by '*O*', for a machine in one planning horizon. Moreover, since each machine component or component has a different age as well as failure characteristics, scheduling preventive maintenance of the entire machine every time will attract high PM costs due to unnecessary PM activities, high penalty costs due to increased downtime as well as engage the limited labour resource needlessly. Moreover, not performing PM on these components at the correct time increases the chances of failure, which on the other hand may lead to even higher CM costs and increased downtime and penalty costs. Thus, optimal scheduling of PM activity on a machine involves identifying the feasible opportunities at which PM needs to be performed for concerned subassemblies which should undergo PM at that opportunity. Furthermore, since the number of labour available for performing maintenance is limited, for the case of PM being scheduled across machines at coinciding PM opportunities, high waiting for maintenance downtime will result, if the combined labour requirement of both the machines exceed the available resources. So, optimally scheduling PM activities for concerned subassemblies at appropriate PM opportunities across all machines is also important to avoid such clashes which cause increased downtime and increased costs. The cost models to calculate the overall production costs and costs associated with each event in the production scenario is presented in the following section.

Similar to section 5, the problem is represented as a string of bits (described in Figure 7.). If m = no. of machines, n = no. of components, o = no. of PM opportunities, then the bit string is of length m*n*o.

Figure 8. Bit string representation of PM schedule for enterprise.

Three algorithms are presented to find the PM schedule which when implemented yields least cost of operation:

6.1. Brute Force Search (BF)

All possible PM schedules are enumerated. The cost of implementing each is evaluated by Monte Carlo simulation and the schedule with the least cost is picked as solution. Consider a case where number of machines is 6, number of PM opportunities per machine is 7 (which evaluates to 7 jobs in two months) and number of components in each machine is 5. The total possible PM schedules are 26*5*7 which is of the order of 1063. This combinatorial explosion, along with the computationally intensive stochastic evaluation of schedule cost makes the run-time intractable. However the result produced by brute force search is optimal, and is useful as target value in analysis of heuristics presented below.

6.2 Memetic Algorithm (MA)

The machine level Memetic Algorithm described in Section 5 is extended to the enterprise level. Instead of using a bit string of length *c*o* (which is the bit string for one machine), the enterprise level Memetic Algorithm uses a bit string of length *m*c*o* as its chromosome. The heuristic function is evaluated for each group of bits of length of *c*o*.

6.3 Particle Swarm Optimization (PSO)

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

The decimal number given by the binary string representation used in the Memetic Algorithm is used as the particle location. Consider the following example of an enterprise with three machines, each machine having 2 components and each schedule having 2 opportunities:

MA Chromosome	10 11	11 10	01 01
PSO particle	11	14	5

6.4. Distributed Algorithm

In the distributed algorithm, the process of preventive maintenance planning is broken down into machine level and enterprise level planning.

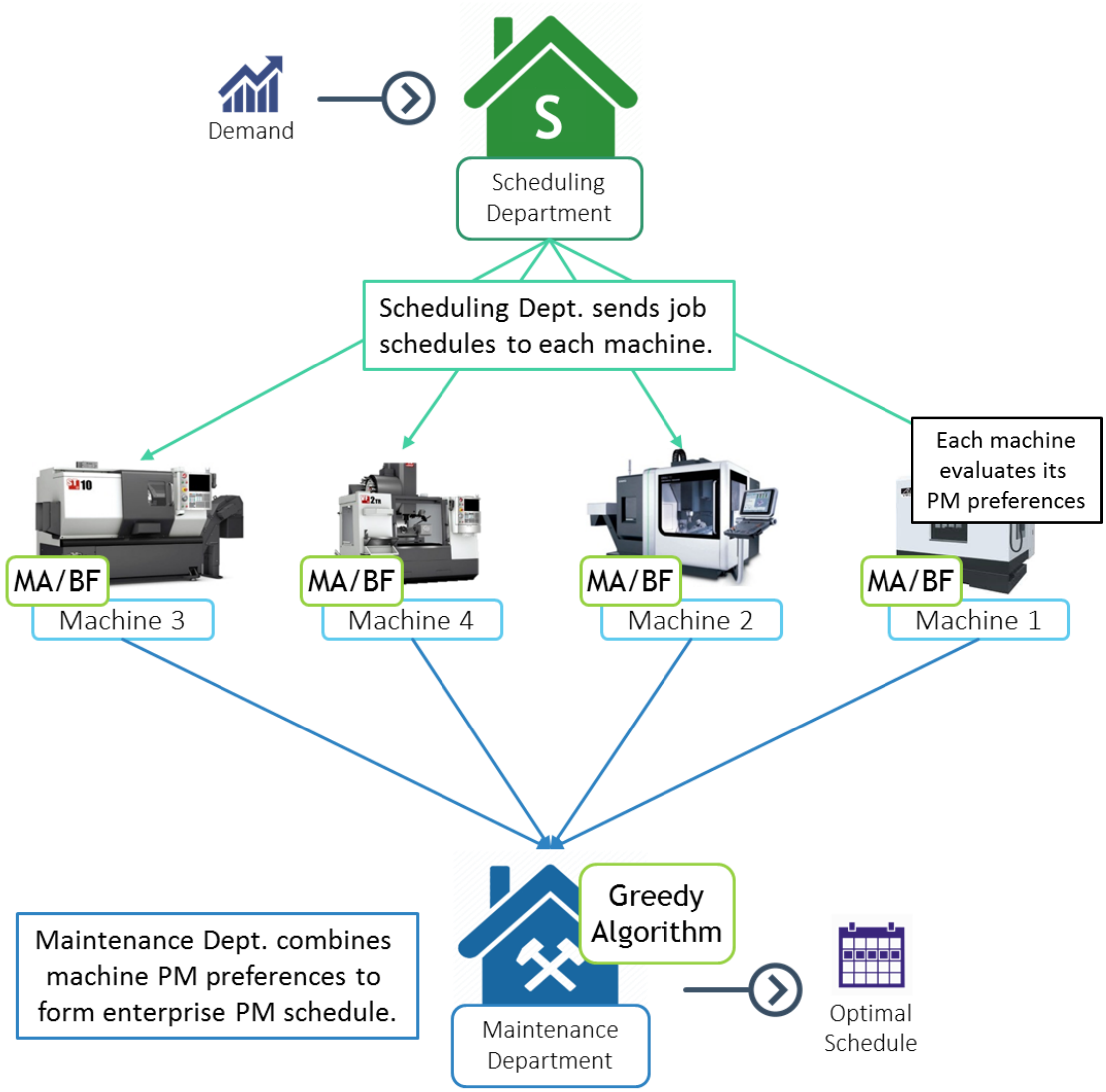


Figure 9. The Two levels of planning of the Distributed Algorithm

6.4.1 Machine Level Planning

Each machine Cyber Twin enumerates all possible PM schedules (including the scenario when no PM is performed) for the job schedule assigned to it, as if it were the only machine in the enterprise. These machine level PM schedules or 'PM preferences' are then simulated multiple times to find the average cost of each. While simulating the PM preferences, it is assumed that labour is always available. The Intensity Factor (IF) is determined for each PM preference as given below. It is a way to compare the cost benefits of two PM preferences of different machines, and is of use to the maintenance department while creating the enterprise level PM schedule.

$$Intensity\ Factor\ (IF)\ for\ a\ schedule\ =\ (Cost\ of\ schedule\ without\ PM) - (Cost\ of\ schedule)$$

```
C ← number of components of machine
N ← number of PM opportunities in job schedule

//enumerate all possible PM preferences
PM_Prefs[] ← all permutations of binary strings of
length C*N

No_PM_Cost = simulate(no PM schedule)

//evaluate cost and IF factor of each PM preference
for each PM_Pref in PM_Prefs:
   PM_Pref_Cost = simulate(PM_Pref)
   PM_Pref_IF = PM_Pref_Cost – No_PM_Cost
```

Machine level planning algorithm.

6.4.2 Enterprise Level Planning

The maintenance department software runs on a separate computer than the machine simulators. It collects PM preferences of all machines and knows the job schedule of each machine. After this, the PM preference with the highest IF is incorporated and all other preferences of those machines, along with those preferences of other machines whose requirement for labour for PM jobs clashes with the incorporated preference, are deleted. CM jobs are not considered while evaluating these labour clashes. Hence there may be some labour clashes due to machine breakdowns while during operation. However, choosing not to consider these breakdowns gives us the benefit of not having to simulate all possible combinations of PM preferences of the machines the enterprise level schedule. This reduces planning time greatly.

```
Collect PM prefs of all machines into a
consolidated list

Sort list in descending order by Intensity
Factor

Repeat till list is empty:
      //Incorporate PM pref with highest IF and
delete it

      //Delete all PM prefs of machine whose
pref was incorporated
      //Delete all PM prefs with labor clash
```

Enterprise level planning algorithm.

6.5. Experiments and Results

The following setup was used for execution and testing of the algorithms:

Intel(R) Core i7-4790 CPU @ 3.60GHz

4GB Memory

Windows 7 Professional 64-bit

JDK 8 for compilation and JRE 8 for execution (Java 8).

No. of Machines (m)	No. of Components (c)	Simulation Count (k)	Algorithm	Run-time	Solution Cost
2	2	1000	Dist	0.59	215341
2	2	1000	BF	286.27	211470
2	2	1000	MA	126.52	219174
2	2	1000	No PM	-	277557
3	2	1000	Dist	0.59	286587
3	2	1000	BF	2058.99	285098
3	2	1000	MA	600.69	308086
3	2	1000	No PM	-	354695
4	2	1000	Dist	0.58	390968
4	2	1000	BF	17453.10	384772
4	2	1000	MA	1042.21	400886
4	2	1000	No PM	-	467434
5	2	1000	Dist	0.60	487249
5	2	1000	BF	36208.67	475638
5	2	1000	MA	1357.52	491535
5	2	1000	No PM	-	549864
6	2	1000	Dist	0.59	600597
6	2	1000	BF	-	-
6	2	1000	MA	1744.55	621392
6	2	1000	No PM	-	712586

Table 6. Comparison of Dist, MA, BF and No PM cost for small problem sizes

The problem size is defined as the product m*c*o where m is the number of machines, c is the number of components per machine and o is the number of PM opportunities (or number of jobs) per machine. 2^{m*c*o} gives the total number of possible PM schedules. Table 6 shows the results of comparison of time and solution cost for small problem sizes for Brute Force Search (BF), Memetic Algorithm (MA) and the Distributed Algorithm (Dist). The cost of not performing any PM (No PM) is calculated using the simulator and shown in the table. It can be used to calculate the cost benefit being obtained from performing preventive maintenance.

Since BF searches the entire solution space, the cost of the solution produced by BF can be considered to be optimal. BF is not a practical approach as its run-time is intractable. It can be seen from the table the run-time increases exponentially. For m=6, the expected run-time is in days and hence the solution is not computed. However since BF solutions are optimal, they are evaluated for small problem sizes to serve as a benchmark for comparing the optimality of solutions produced by other algorithms.

It is clear from Table 6. that the run-time of the Distributed Algorithm is very small for small problem sizes, whereas the solution produced is nearly the same as that of BF. Moreover, Dist always performs better than MA in terms of run-time as well as solution cost. The comparison of solution costs of the three techniques along with no PM cost is done in Figure 10. The variation in the values of the optimized costs for each case shows the scope to which the cost could be reduced. It can be observed that BF produces the most optimal solution, as expected from its computational technique.

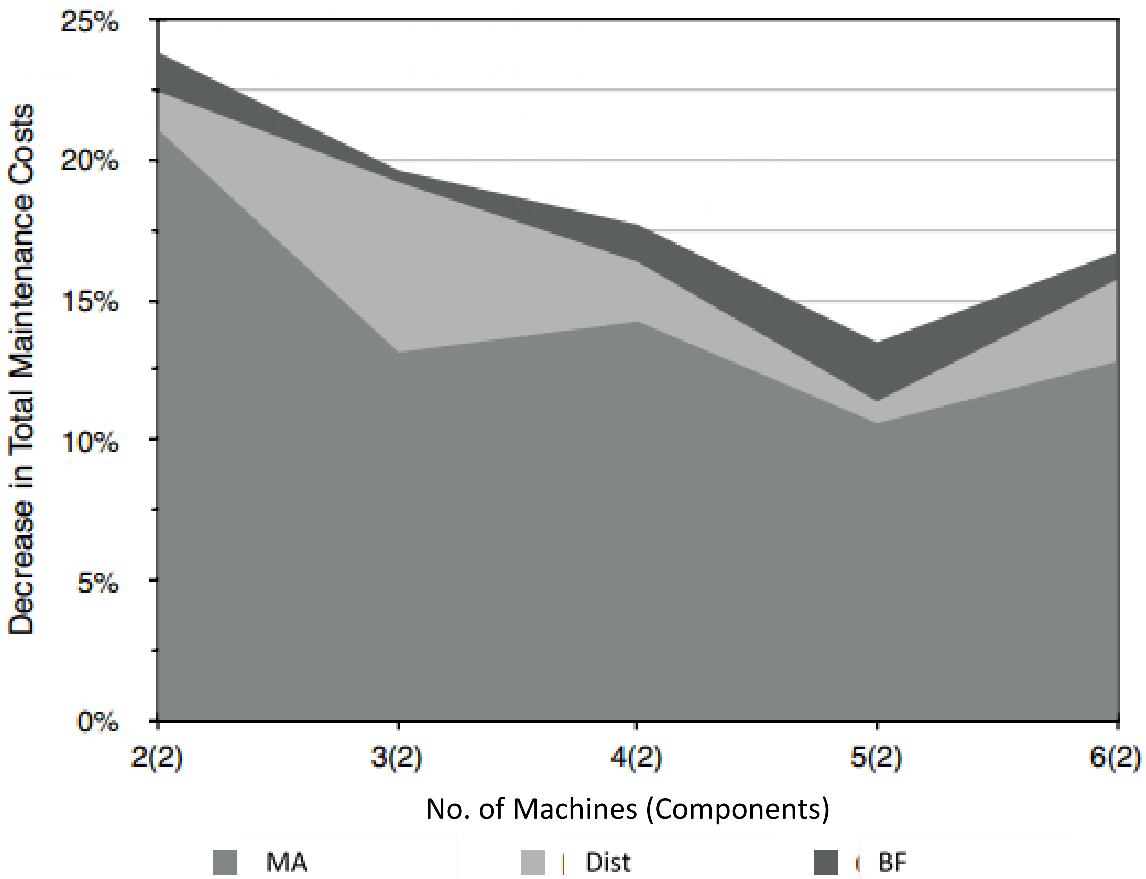


Figure 10. Comparison of Dist, BF and MA with No PM cost for small problem sizes

Table 7 presents the results of comparison of Memetic Algorithm (MA), Particle Swarm Optimization (PSO) and the Distributed Algorithm (Dist). The cost of not performing PM is calculated using the simulator and shown as a benchmark for scope of improvement due to PM planning in each case. The solutions produced by PSO are close to that produced by Dist and MA. However the run-time is in thousands of seconds for simulation count of just 100 (whereas MA and Dist use a simulation count of 1000). Hence PSO approach is rejected as the run-time as not as small as desirable.

No. of machines	No. of components	Algorithm	Simulation Count	Run-time	Solution Cost
6	3	Dist	1000	125.00	833786
6	3	MA	1000	1,809.31	869417
6	3	PSO	100	2,432.87	847886
6	3	No PM	1000	-	957022
7	3	Dist	1000	170.59	1029075
7	3	MA	1000	2,151.81	1074340
7	3	PSO	100	2,950.96	1015936
7	3	No PM	1000	-	1208880
8	3	Dist	1000	151.06	1183031
8	3	MA	100	3,383.61	1183398
8	3	PSO	1000	2,362.33	1237933
8	3	No PM	1000	-	1289824
6	10	Dist	1000	277.86	2137097
6	10	MA	1000	2,063.15	2211524
6	10	No PM	1000	-	2263155
7	10	Dist	1000	260.77	2525412
7	10	MA	1000	2,460.27	2584042
7	10	No PM	1000	-	2666506
8	10	Dist	1000	249.38	2877453
8	10	MA	1000	2,824.10	2913199
8	10	No PM	1000	-	3026668

Table 7. Comparison of Distributed Algorithm (Dist), Memetic Algorithm (MA) and Particle Swarm Optimization (PSO)

The Distributed Algorithm produces solutions better than the Memetic Algorithm, though in some cases both the solutions are very close. However the run-time of MA is nearly 10 times or more than that of the Distributed Algorithm. The run-time of the Distributed Algorithm is only a few seconds and does not vary much with the problem size. It is also to be noted that in case of smaller problem sizes (Table X), the machine preferences in the distributed algorithm are calculated using (machine level) Brute Force Search and those for larger problem sizes (Table Y) are calculated using the Memetic Algorithm. For smaller problem sizes, the use of machine level BF is considerably faster than MA, as the MA has a relatively large overhead to initializing and evolving generations. This is why there are two different orders of run-times for the distributed algorithm in the two tables- a fraction of a second and 10s of seconds.

To summarize, it can be said that the cost of solutions of the distributed algorithm is close to optimal, i.e. the cost of solution produced by Brute Force Search. Though the MA and PSO may produce slightly better solutions than the Distributed Algorithm at times, the latter is the clear winner in terms of run-time. Such a small run-time can bear huge significance, as it will enable an extensive What-If analysis in the enterprise that can be of great use in determining optimal operation parameters. This is not possible for MA and PSO.

A test was conducted with a configuration of 27 machines, each with 8 components for which the solution was produced in 297.45 seconds. With each additional machine, there is additional computational resource being added (as the Cyber Twin of each machine is run on a separate PC). Hence the machine level computation is unaffected by the number of machines in the enterprise. The enterprise level computation is not computationally intensive (simply greedy algorithm), and does not play a bottleneck role in algorithm complexity. Hence the Distributed Algorithm has infinite scalability in number of machines.

7. Dynamic Planning

First, we discussed a simulator for machines, and used this in Monte Carlo simulation for machine level optimization. Later, decentralized planning was performed for the enterprise by interconnecting constituent Cyber Twins. However, the predictions being made by the stochastic model are not completely accurate. Uncertain events may still occur and disrupt plans made using the above techniques. Hence a robust method is needed that can modify plans and schedules when uncertain events occur. Dynamic planning is a proposed real-time solutions that can do just this.

A distributed software system is developed that runs in sync with the enterprise, the Cyber Twins monitoring asset states in real-time. Whenever an event occurs, the system determines the best action to take and makes adjustments to operation plans. Depending upon the level of sophistication of technology and desired level of automation, the decision can be implemented completely automatically, or appear as a recommendation to a manager who can select the course of action remotely through a smartphone application. In case of Computer Numerical Control (CNC) machines, sensing the event and responding to it can be instant.

Since the system runs in real-time, it is important that all planning and re-planning actions are instant. The machine and enterprise level planning and optimization algorithms discussed above have a run-time of a few seconds to a few minutes, and hence are fit for use. Whenever an uncertain event occurs, the Cyber Twin states and constraints are updated and the enterprise level planning algorithm is run again. A new schedule that takes the event into account is obtained and conveyed to all Cyber Twins.

The most novel features and improvements over existing works are in terms of efficiency (lesser time for operations planning) and flexibility (quick response to change in operational parameters).

With this, the complexity of the problem increases by a notch. The system must handle information in real-time and make the best use of it. A flexible communication mechanism to handle the continually increasing complexity and variability in the planning processes in devised. The solution is provided by developing a communication network in which Cyber Twins are social and share information with only those Cyber Twins for whom it may be relevant.

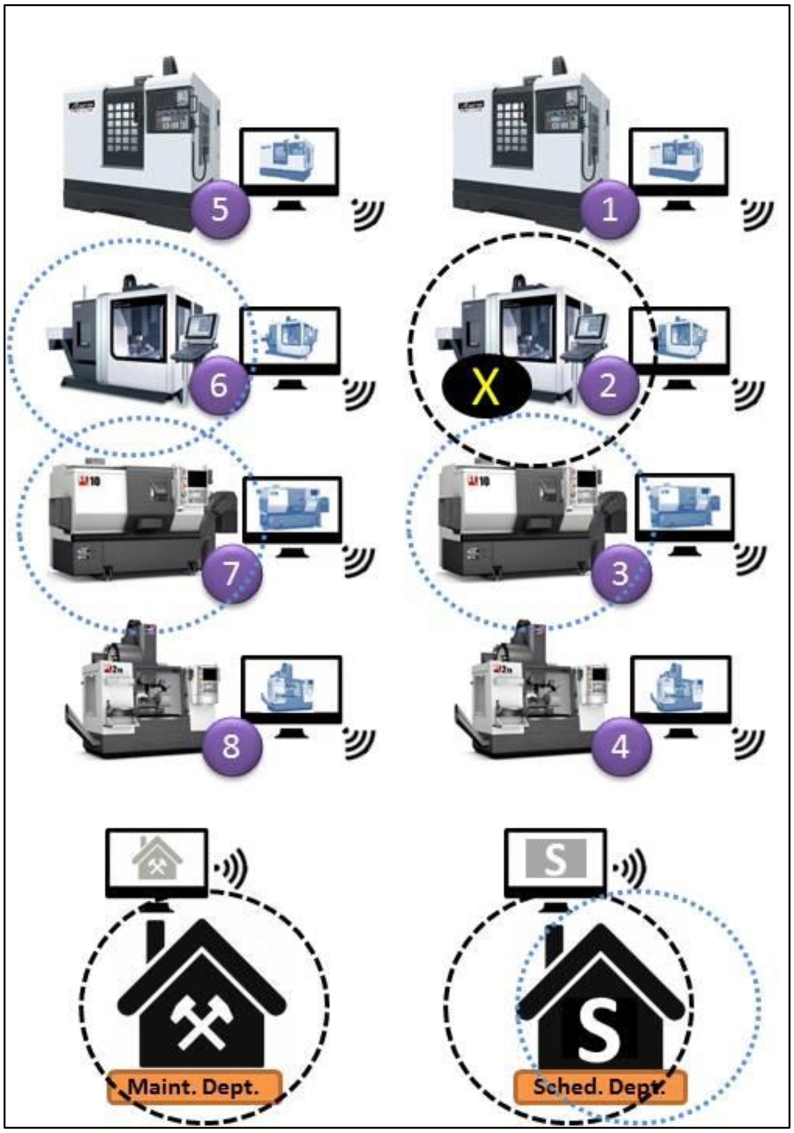


Figure 11. Formation of groups during re-planning

Consider the following (simplified) example shown in Figure 11. The enterprise has 8 machines, each can produce different types of jobs. Machine 2, that can produce jobs types A and B breaks down. It forms a group with the Maintenance Dept. and Scheduling Dept. to share updates on its state. The Maintenance Dept. takes actions towards arranging labour for repair so that normal operations can be restored. The Scheduling Dept. works towards readjusting job schedule so that delays are minimized. For this purpose it forms another group with Machines 2, 3, 6 and 7 as they can all produce job types A and B. In this group, they negotiate the readjustment of jobs, and the new job schedule is generated. Thereafter, the PM planning is done as according to the Distributed Algorithm described in Section 6.

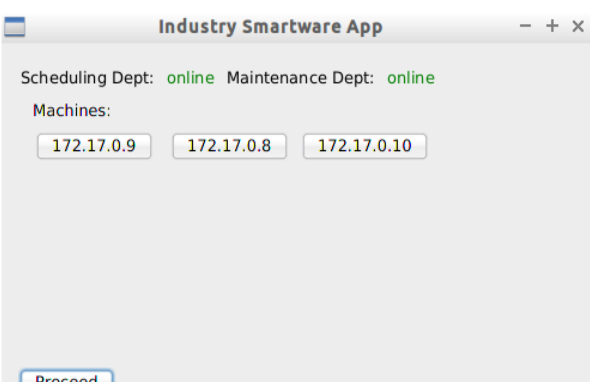
8. Software

Industry ready software has been developed during the course of the project. The software system for Enterprise Level planning and Dynamic Planning is a distributed software developed entirely in Java. For the simulation, each function of the Cyber Twin is coded as an independent Thread, with a central Thread managing the control flow. This makes it easy to add new functionality. Each Cyber Twin stores the data of events in a SQLite database. This serves the purpose of logging, as well as providing data for scientific analysis during operations planning and updating the simulation model of Cyber Twin. All networking has been implemented using TCP sockets. The use of multicast sockets is done for discovery of Cyber Twins during software. Each Cyber Twin is expected to the run on a different PC.

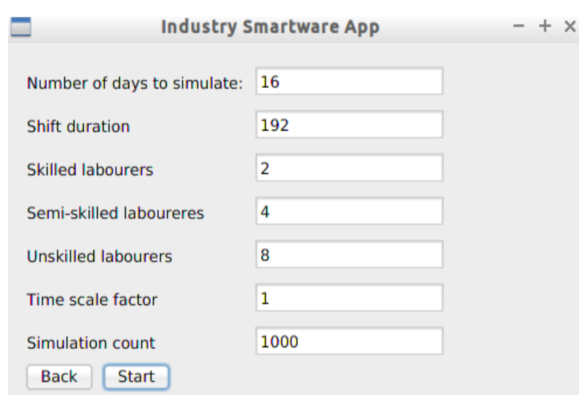
A Central Monitoring system is provided that can serve administrative and managerial purposes. It is also expected to be run on a separate PC. All the Desktop applications are developed in Java and distributed as executable JAR files. Hence they are platform independent. The size of the JAR files is less than 10MB, and can even be run easily on embedded systems that support the Java Runtime Environment.

A webapp is developed to serve as the interface between Cyber Twins and physical assets of the enterprise. The operator of the physical asset is expected to make use of it to report events related to the asset. It can also be used for remote monitoring. The webapp is responsive and can be opened on any desktop or mobile browser. This makes it usable on PCs, laptops, smartphones, tablets etc. It is hosted through a lightweight Nano-HTTP Java server, being run as a part of the Central Monitoring System, and hence requires no special setup of server.

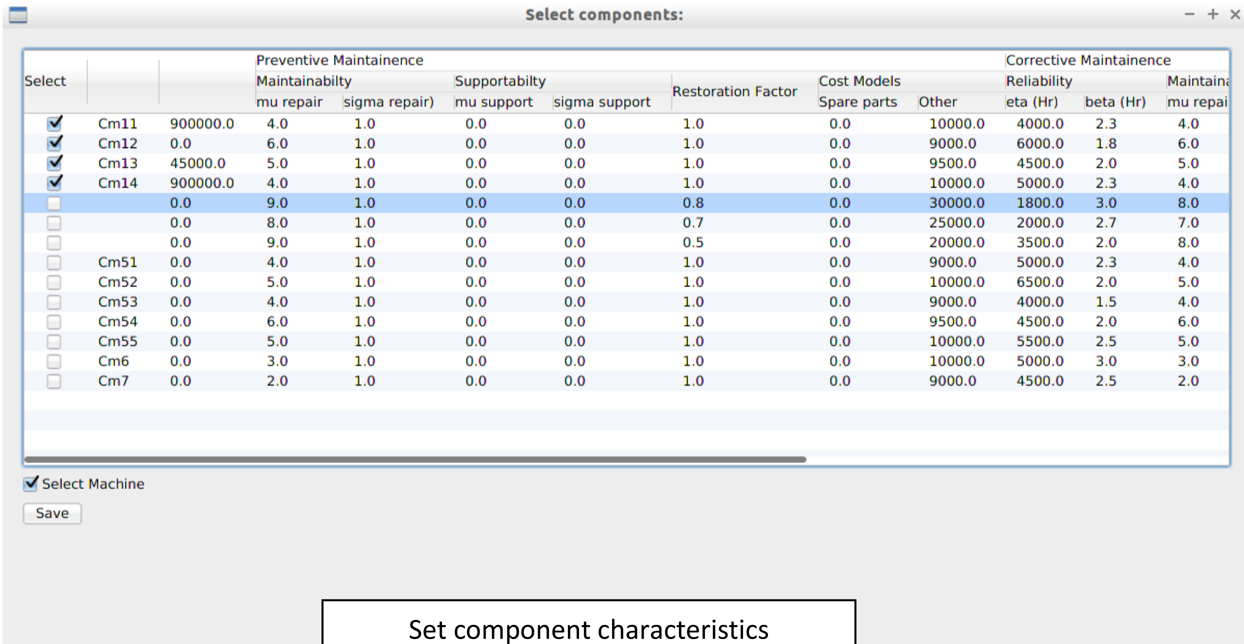
Screenshots of various display windows of the developed software are shown below.



Waiting for entities to come online.

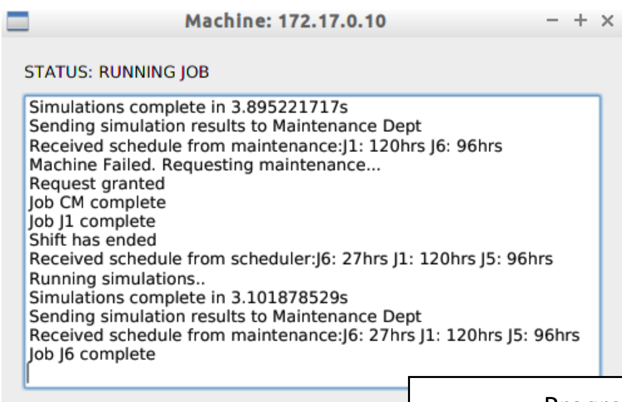


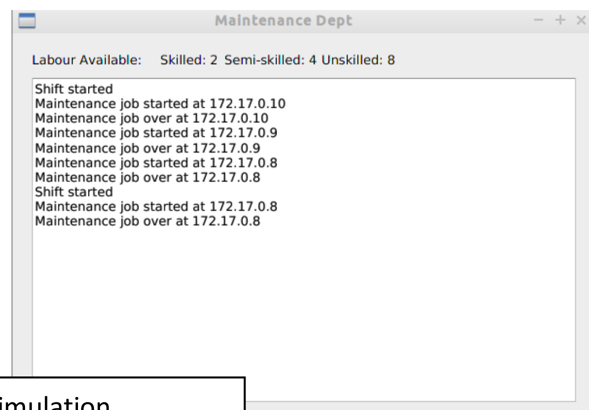
Set simulation parameters



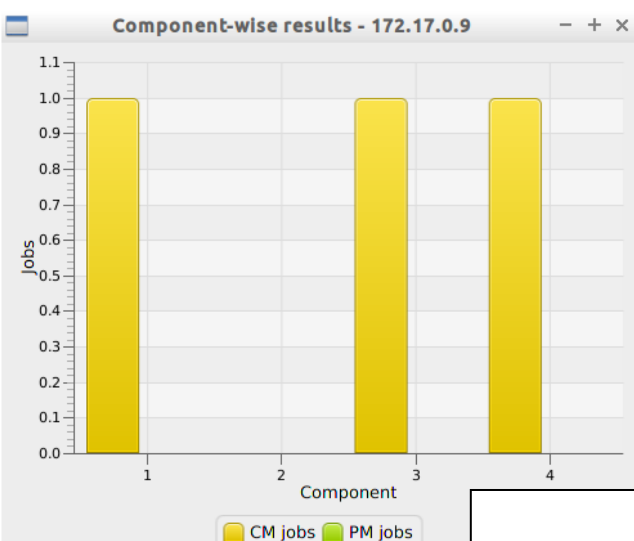
Select			Preventive Maintainence: Maintainabilty: mu repair	sigma repair)	Supportabilty: mu support	sigma support	Restoration Factor	Cost Models: Spare parts	Other	Corrective Maintainence: Reliability: eta (Hr)	beta (Hr)	Maintain: mu repai
☑	Cm11	900000.0	4.0	1.0	0.0	0.0	1.0	0.0	10000.0	4000.0	2.3	4.0
☑	Cm12	0.0	6.0	1.0	0.0	0.0	1.0	0.0	9000.0	6000.0	1.8	6.0
☑	Cm13	45000.0	5.0	1.0	0.0	0.0	1.0	0.0	9500.0	4500.0	2.0	5.0
☑	Cm14	900000.0	4.0	1.0	0.0	0.0	1.0	0.0	10000.0	5000.0	2.3	4.0
☐		0.0	9.0	1.0	0.0	0.0	0.8	0.0	30000.0	1800.0	3.0	8.0
☐		0.0	8.0	1.0	0.0	0.0	0.7	0.0	25000.0	2000.0	2.7	7.0
☐		0.0	9.0	1.0	0.0	0.0	0.5	0.0	20000.0	3500.0	2.0	8.0
☐	Cm51	0.0	4.0	1.0	0.0	0.0	1.0	0.0	9000.0	5000.0	2.3	4.0
☐	Cm52	0.0	5.0	1.0	0.0	0.0	1.0	0.0	10000.0	6500.0	2.0	5.0
☐	Cm53	0.0	4.0	1.0	0.0	0.0	1.0	0.0	9000.0	4000.0	1.5	4.0
☐	Cm54	0.0	6.0	1.0	0.0	0.0	1.0	0.0	9500.0	4500.0	2.0	6.0
☐	Cm55	0.0	5.0	1.0	0.0	0.0	1.0	0.0	10000.0	5500.0	2.5	5.0
☐	Cm6	0.0	3.0	1.0	0.0	0.0	1.0	0.0	10000.0	5000.0	3.0	3.0
☐	Cm7	0.0	2.0	1.0	0.0	0.0	1.0	0.0	9000.0	4500.0	2.5	2.0

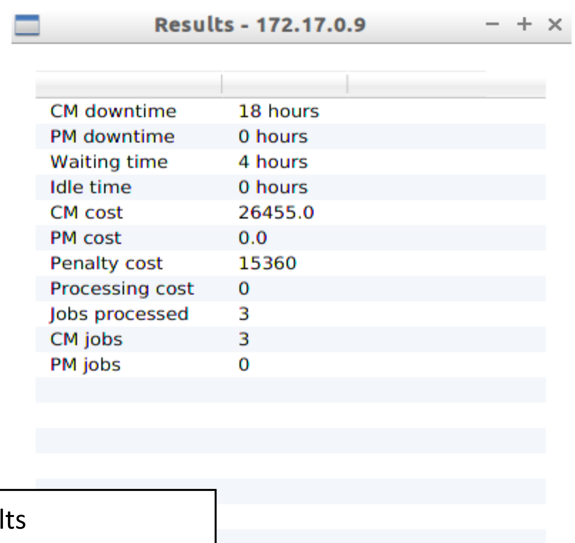
Set component characteristics





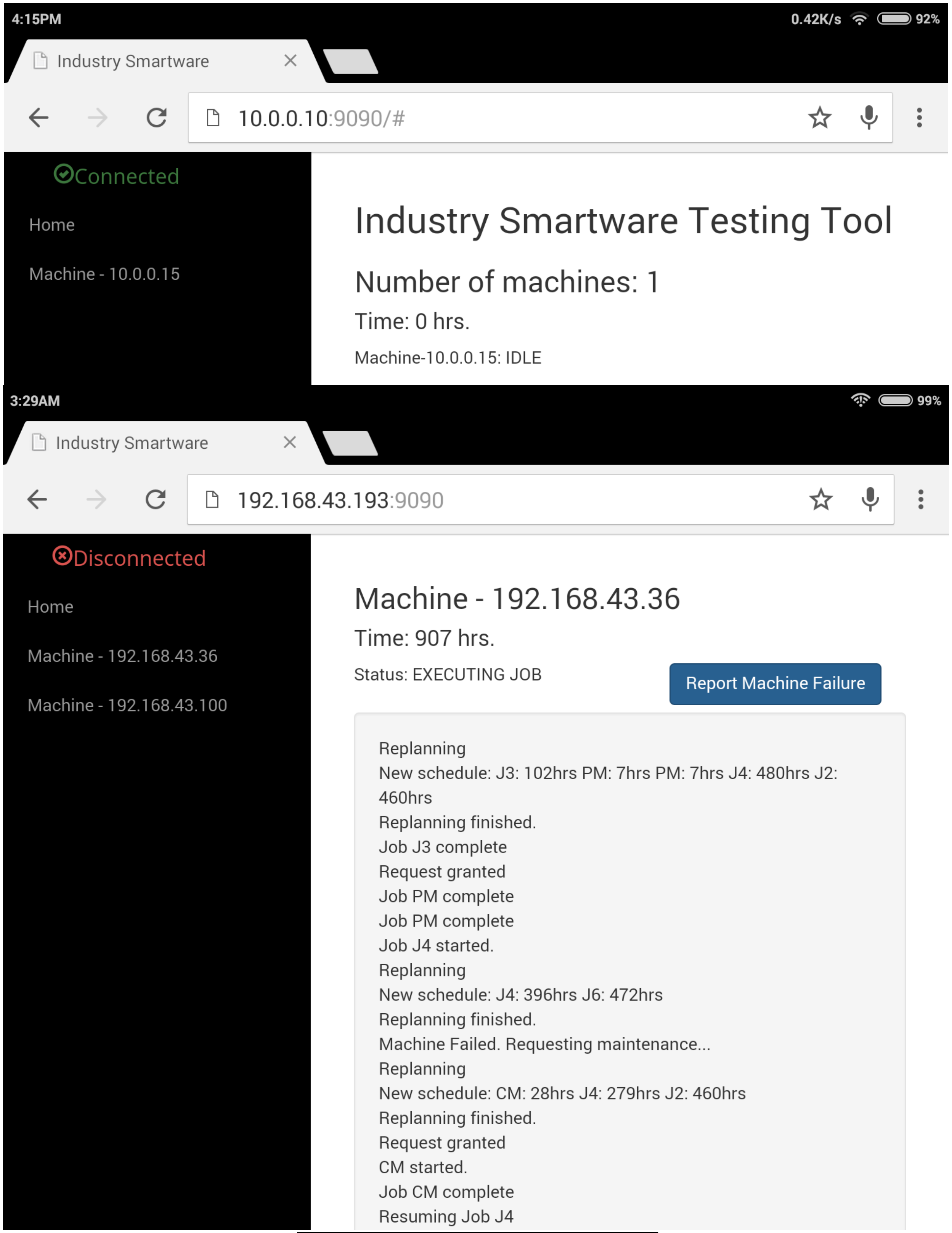
Progress of simulation





CM downtime	18 hours
PM downtime	0 hours
Waiting time	4 hours
Idle time	0 hours
CM cost	26455.0
PM cost	0.0
Penalty cost	15360
Processing cost	0
Jobs processed	3
CM jobs	3
PM jobs	0

Results



Web App to monitor and report machine status

9. Conclusion

In conclusion, a system is developed that is an effort towards introduction of Smart Manufacturing to an industries. The system allows for a varying degree of automation which is commensurate to monetary investment and technical upgrades (sensors, actuators, CNC machines). Most importantly, it allows for a low cost and easy implementation that can be useful to industries in realizing the scope of Smart Manufacturing techniques, by installation in their industry.

On a technical note, work is done to develop and implement machine simulation models. A multi-component machine is considered in the work, which is a detailed model and is expected to yield results with better accuracy. The concept of Cyber Twins leverages such a model along with concepts of Cyber Physical Systems and creates a virtual replica of physical assets of an enterprise. These Cyber Twins are intelligent and social, and are fundamental elements of the system. The project is in collaboration with AVTEC Ltd, Pithampur and a Cyber Twin for a real machine at their plant has been developed.

The enterprise is modelled as a multi-agent system of Cyber Twins. A philosophy of decentralization is used wherever applicable, which aides greatly in reducing design complexity. A distributed algorithm is developed to leverage the concept of Cyber Twins and provide a fast and scalable solution to the decision variable of preventive maintenance planning. Finally, a novel approach of Dynamic Planning is developed that adds robustness to the system by making it capable of effectively handling uncertain events such as machine failures. A communication network of Cyber Twins has been leveraged for this purpose that streamlines the flow of information in the industry. Industry-ready software is developed, that is lightweight and platform independent.

A publication- Upasani, K., Bakshi, M., Pandhare, V., Lad, B.K.,*"Memetic Algorithm to Optimize Preventive Maintenance Schedule for a Multi-component Machine"* has been submitted to the International Journal of Performability Engineering, and is currently under review. A journal paper on the Distributed Algorithm, and a patent application for the Dynamic Planning System are in the final stages of submission.

10. References

1. Forget P, D'Amours S, Frayret JM. Multi-behavior agent model for planning in supply chains: An application to the lumber industry. 2007.
2. Frayret JM, D'Amours S, Montreuil B. Coordination and control in distributed and agent-based manufacturing systems. Prod Plan Control 2004;15(1):42–54
3. P. Derler, E. A. Lee, A. Sangiovanni-Vincentelli. Modeling Cyber-Physical Systems, January 2012.
4. Edmonds B, Meyer R. Simulating Social Complexity: A Handbook.
5. Wipro - Smart Manufacturing. http://www.wipro.com/industries/manufacturing/services/smart-manufacturing/
6. Stadtler H., 2004. Supply chain management and advanced planning—basics, overview and challenges.
7. Anthony, 1965; Hax and Meal. Hierarchical Production Planning. 1975.
8. Yang Z., Djurdjanovic D, Ni J. Maintenance scheduling in manufacturing systems based on predicted machine degradation. 2007.
9. Pravin P. Tambe & Satish Mohite & Makarand S. Kulkarni. Optimisation of opportunistic maintenance of a multi-component system considering the effect of failures on quality and production schedule: A case study. 2013.
10. Caridi M, Cavalieri S. Multi-agent systems in production planning and control: an overview. Prod Plan Control 2004;15(2):106–18
11. Russel S, Norvig P. Artificial intelligence: a modern approach. 2nd ed. Englewood Cliffs, NJ: Prentice-Hall; 2003.
12. Wooldridge, M. (2009) An Introduction to MultiAgent Systems, John-Wiley and Sons.
13. Gentle JE. Random Number Generation and Monte Carlo Methods. 2013.
14. Kijima, M. "Some results for repairable systems with general repair," Journal of Applied Probability, 20, 851-859, 1989.
15. A.I. Anosike, D.Z. Zhang, An agent-based approach for integrating manufacturing operations,2006.
16. Luping Zhang, T.N. Wong, Sicheng Zhang, S.Y. Wan, "A multi-agent system architecture for integrated process planning and scheduling with meta-heuristics".

17. ‘Industrie 4.0 - Smart Manufacturing for the Future’, Germany Trade & Invest, Federal Republic of Germany.
18. *Draft Policy on Internet of Things, Department of Electronics & Information Technology (Deity), Ministry of Communication and Information Technology, Government of India. Link: https://mygov.in/sites/default/files/master_image/Revised-Draft-IoT-Policy-2.pdf).*