Adaptive Genetic Algorithm for Production System

تحسين أداء خوارزمية الخريطة الجينية قي أنظمة الإنتاج

Prepared by: Majdi Jalal Al-Saaideh

Supervised by: Dr. Akram M.O. Al-Mashaykhy

This Thesis is submitted in Partial Fulfillment of the Requirements for the Master's Degree of Science in Computer Science

College of Computer Science and Information

Amman Arab University

(Aug, 2013)

Acknowledgement

قال تعالى في محكم كتابه :"لئن شكرتم لازيدنكم" ويقول رسولنا الكريم:"من لم يشكر الناس لا يشكر الله".

لذا فأنني أجد من الواجب علي ان اشكر كل من ساعدني في انجاز هذا العمل ،في كافة مراحله منذ البداية حتى وصل الى ما بين ايديكم و وقد استغرق هذا العمل من الجهد اكثر من سنة كاملة من الجد والعناء .

واخص بالذكر الدكتور اكرم عثمان المشايخي

والذي كان له البصمة الواضحة في هذا العمل من توجيه واعطائي الافكار والحقائق العلمية واسلوب البحث حيث اشرف بنفسه على هذا العمل بكافة مراحله وما هذا العمل الا ثمرة جهد مشترك معه فله مني كل الشكر والثناء.

وكذلك الدكتور علاء الحمامي عميد كلية تكنولوجيا المعلومات في جامعة عمان العربية للدراسات العليا له منا كل الاحترام والتقدير.

واخيرا وليس اخرا اهدي هذا العمل الى والداي العزيزين الذين سهرا الليالي معي اثناء الاعداد لهذا العمل .

Authorization of Dissemination

I, the undersigned "Majdi Jalal Abd_Alhadi Al_Saaideh" authorize here by "Amman Arab University for Graduate Studies" to provide copies of this dissertation to libraries ,institutions, agencies, and any other parties upon their request.

Name: Majdi Jalal Abd_Alhadi Al_Saaideh

Signature:

Resolution of the Examining Committee

This thesis titled: "**Adaptive Genetic Algorithm for Production System**". Has been defended and approved on / 23 / 8 /2013.

Examining Committee	Signature
Prof. Dr. Muzhir S. Al-Ani	
Dr. Akram M. O. Al-Mashaykhi	
Dr .Feras Al-Mashakbah	

ABSTRACT

In recent years, with the great competitive in the market place, many researchers proposed methods and algorithms that aim to find adaptive optimization strategies to solve the problem of planning and scheduling in manufacturing systems.

Such kind of studies fall under the list of artificial intelligence and interest in the acquisition of the knowledge of God, and enable them to make and implement decisions on behalf of the human.

Here is evident obvious difference between artificial intelligence and object-oriented programming, where the latter does not have the ability to make decisions alone should be available directs the user to what you should do its work.

The idea thesis of difference in views between officials of the production companies, as some of them had to give the machine a specific type of products for the completion of the process addressed first and start another type, while the other section to distribute more than one product on more than one machine where the processed productby special machine which, where this method helps us to achieve the exploitation and optimum utilization of available resources in the factory.

It is the principle of achieving major benefit and the parable of the exploitation of resources have put forward in our project several methodologies(on the shelf), It helps a genetic algorithm to find the ideal distribution of products via the available resources in a random manner, based on the equations, especially in genetic algorithmwhere the results showed a significant improvement

compared to when applying the algorithm before the amendment, when you return to the improvement for the cost, we will find it reached the amount of 25%, while the improvement in terms of stability for the production lines is about 73%, either in terms of time used to handle a particular product was improved approximately 36 % Finally, the size of the representation of the product has been reduced to the simplest form to be treated and it have reached 30% than it was when the application of genetic algorithm.

Arabic summary

تحسين أداء خوارزمية الخريطة الجينية قي أنظمة الإنتاج

يعتبر الذكاء الاصطناعي احد اهم واحدث الفروع المنبثقة من علم الحاسوب ،والتي تهتم بان يكون للحاسوب قدرة على اكتساب المعرفة من البيئة المحيطة واتخاذ قرار معين بناء على ذلك، فاذا استطعنا ان نجعل الحاسوب والانظمة المحوسبة تحل مكان الانسان بحيث تكون قادرة على اتخاذ القرارات نيابة عنه، في هذا الحالة نستطيع وصف النظام بانه نظام ذكي .

وفي النظر الى مشروعنا في هذه الرسالة ، فاننا نعمل على ايجاد خوارزمية معدلة تساعدنا على الحصول على التوزيع المثالي لمنتجات معينة عبر الموارد المتوفرة في مصنع معين او مؤسسة ما ، بحيث تكون نسبة الاستغلال والاستفادة من هذه الموارد افضل واكبر من ما كانت عليه عند تطبيق الخوارزمية قبل التعديل،مع الاخذ بعين الاعتبار انه يجب تسليم جميع المنتجات المطلوب معالجتها بوقتها المحدد دون تاخير ، وكذلك ان تتم معالجة كل منتج عبر الالة الخاصة فيه وكذلك يجب الاخذ بعين الاعتبار معالجة المنتجات الطارئة في وقتها المحدد ودون التسبب في ارباك نظام الانتاج.

الاهداف المرجوة من انجاز هذا العمل :

يعطي هذا التطبيق رؤية بعيدة وشاملة لصاحب العمل عن كيفية توزيع المنتوجات عبر المعدات خلال مدة زمنية كبيرة وبالتالي تتكون لديه معرفة عامة عن كيفية سير العمل واحتياجات العمل المستقبلية.

هذا التطبيق يمّكن صاحب العمل من تقليص عدد الموظفين داخل مؤسسته الانتاجية مثل مهندسي الانتاج ،وبالتالي ينعكس ذلك انعاكسا ايجابيا على التكلفة بالنسبة لصاحب العمل .

نسبة الخطأ في القرارات الصادرة عن التطبيقات المبرمجة عادة ما تكون نسبة الخطأ فيها اقل من القرارات الصادرة عن الانسان (مهندسي الانتاج).

حيث اظهرت نتائج الدراسة تطورا واضح وملموس وعلى مختلف المقاييس المستخدمة في هذه الدراسة ،فبالنظر الى التكلفة كان قد وصل مقدار التحسن الى ما يقارب 25%،بينما كان مقدار الاستقرار في نظام

التوزيع المستخدم اعلى منه مما كان عليه بحوالي 73%، اما على صعيد الوقت المستخدم لانجاز العمل فقد اظهرت النتائج التحسن بمقدار 36% ،واخيرا حجم التمثيل للمنتج المراد معالجته تم تقليصه لشكل ابسط وذلك بمقدار ما يقارب 30%.

List of Contacts

List of Figures

List of Tables

Abbreviations

AI……………………….....……………………………………...Artifici
al Intelligent

CAMPS_MP………....Constraint Airlift Mission Planning
Sheduler_Mission planner

CEP……………………………………………………………….Child
End Process

CRP……………………………………………………...Capacity
Requirement Planning

DAI……………………………………………………...Distributed
Artificial Intelligent

DPS……………………………………………………..….Distributed
Problem Solving

ELSP……………………………………………......Economic Lot
Scheduling Problem

EPT………………………………………………………………..End
Process Time

FMS……………………………………………………...Flexible
Manufacturing System

GA………………………………………………………………Gene
tic Algorithm

IEFPSS…………...Integrated Efficient and Flexible Planning and Scheduling System

IP3S………………………...…Integrated Process-Planning-Production Scheduling

JADE………………………………………..Java Agent Development Environment

MAS………………………………………………………......Multi Agent System

MCS…………………………………………………………...Machine Current Slot

MRP……………………………………………..….Material Requirement Planning

MTO……………………………………………………………....Make To Order

MTS…………………………………………………………….....Make To Stock

OSSP……………………………………………….Open Shop Scheduling Problem

PCB…………………………………………………………....Printed Circuit Board

PGA…………………………………………………...…Parallel Genetic Algorithm

POMDPS…………………………….....Partially Observable Markov Decision Process

PRSA……………………………………Parallel Recombinative Simulated Annealing

RCCP…………………………………………………Rough Cut Capacity Planning

TRIPS…………………………………...……The Rocheste Interactive Planning System

Chapter 1

Introduction

Agent-oriented problem solving strategy gained a high interest last years. This strategy deploys several various entities to solve the specified problem.

Entity (Agent) has special characteristics such as: autonomy, heterogeneity, complexity and others. However, they may communicate with each other by means of communication.

Distributed Artificial Intelligence (DAI) is the field in which systems are designed to have intelligence distributed over a number of distributed nodes or agents (Wan 1996)[44].

An intelligent agent distinguished by having knowledge about the problem space and howto solve this problem. This intelligence is very useful when the problem under consideration is intrinsically distributed. DAI system can generally be designed from two perspectives: Distributed Problem Solving (DPS) or Multi Agent System MAS.

When considering how the work of solving a particular problem can be divided among the different cooperated agents, then we are talking about DPS. These agents may share the knowledge about the problem and about the developing solution. On the other hand, MAS system composed of a number of autonomous agents who are able to communicate and collaborate with each other to achieve common goals.

Mathijs (Mathijs de Weerdt 2005) classifies agents into two categories according to the techniques they employ in their decision making: reactive agents, who base their next decision on their current sensory input, and planning agents who take into accountanticipated future developments[35]. Clearly, a planning agent is expected to come up with an optimal/shortest or near-optimal solution in most cases, especially in an environment full of dynamicity and uncertainty such as our proposed one.

Along a different dimension, agents may be organized in two different ways:

Centralized and Decentralized.

In centralized planning environment, goals, rules, constraints, and resources from individual agents are accumulated at a central place and a centralized planner is used to generate a global schedule. In contrast, in decentralized planning each agent generates and maintains its own plan[34].

Properties of the multi-agent system have a significant impact on the solution method that is to be chosen to solve any multi-agent problem. Such properties for example are thatthe communication in most multi-agent systems is limited, and each agent has its goals, and these agents may somehow depend on each other.

Our problem is addressing a centralized multi-agent system that consists of several autonomous production planning agents. These agents are distributed to solve the problem of finding the appropriate plan/schedule combination for the production system problem without violating any of the production system rules or constraints. Agents are managed by a central coordinator.

You must acknowledge that we have more than one option to control the system either through an agent which is the first choice for us in case we faced difficulty in applying the system by intelligent agent we will work to create a system simulation theory as an option second or create an integrated system but are controlled by tools,It is not intelligent agent.

“Planning refers to the generation of activities that satisfy a current set of goals while scheduling is the association of these specific activities with particular times and resources while satisfying specified constraints”, (S.Das 1999)[15].

“Planning and scheduling involve determining when to perform which activities as the production system capabilities (e.g. machines capacity)”, (Bradley J. Clement 2002)[6].

The problem is to produce a plan by each agent with maximal expected return, given the following domain information:

1- A set of autonomous agents with their assumed capabilities.

2-A set of the available machines with their assumed setup times, capacity and the sub products that each one produce.

3-A production structure that shows all products and sub products produced by the manufacture and their precedence relations.

4-A set of orders arrived at random time slots. For each order, the quantity and the dead line for delivery must be determined in advance.

5-A set of initial conditions, which describes the system current state.

Planning and Scheduling problem is considered as NP-Hard problem(the problem of finding an optimal feasible allocation ,where all tasks meet physical and timing constraints), in which a perfect solution cannot be achieved and one solution cannot be proved to be the best of the others in all situations. Because of this complexity of our case study, evolutionary Algorithms- as an optimization technique- are suitable to deal with these kinds of problems because they are considered probabilistic search algorithms and efficient to search a large and complex space of solutions to find a nearly optimal one. Thus, we aim to use the Genetic Algorithm (Goldberg, 1989) and (Mitchell,1997) which was introduced by John Holland in 1970 (Holland,1975), as one of the well-knownevolutionary algorithms to plan and schedule the production system, thus achieving near optimal results[21].

“The traditional manufacturing planning process is divided into 7 planning modules: production planning, resource planning, master scheduling, rough-cut capacityplanning (RCCP), material requirements planning (MRP), capacity requirements planning (CRP), and detailed scheduling”,(Edward F. Watson 1997)[18].

In today’s high competition between various production systems, Production planning becomes a pivotal task. Discovering an efficient and flexible production system planning and scheduling strategy will result in labor and work-in process inventory cost reduction due to the fact that an efficient production system makes an efficient resources (machine) utilization and can deliver orders on-time while maintaining all system constraints (rules on machines, products and other system components).

The N-job, M-machine flowshop problem is a special case of the general production planning and scheduling problem, called the jobshop problem. The problem specification states that all jobs shall flow between the machines in the same order. The solution concerns finding job sequence for all machines that optimize a given objective measure, usually a function of processing times.

The Open-Shop Scheduling Problem (OSSP) is also a complex and common industrial problem. OSSP states that there is a set of operations. These operations need to be performed on one machine or more. The question here is how to find an efficient method to optimize the schedule of these operations on the existed resources in terms of the time slot when all these requested operations finished execution, (H-L Fang 1994)[23].

Thus, production planning concerns the generation of a sequence of production tasks for longer periods of time, given the products, processing times and machines required to produce each one and orders demands. In contrast, scheduling means the assignment of resources to activities and the determination of starting respectively ending times for the execution over a short period of time.

Our work is concerned with delivering an Integrated Efficient and Flexible Planning and Scheduling System (IEFPSS) for Multi-Agent Production System Using the Genetic Algorithm.

1.2. Statement of problem

In factories, which contains production lines usually gets poor distribution of Orders on these lines so as to many reasons, including poor performance for engineers production responsible for the distribution or otherwise sometimes that there are production lines proportion of utilization is very high compared with other productionlines,here we will try to find a particular strategygive the best possible distribution between these lines, based on the equations in production Systems.

1.3. Refinement and Extended Planning

“The Process planning is the systematic determination of the detailed methods by which parts can be manufactured from raw material to finished products” (Smed 2003)[40].

The notion of a plan is very general and encompasses several types of problems such as path planning, production planning. Essentially, a plan is a (partially) order set of actions that aims to achieve a certain goal, (Mathijs 2003)[35].

A flexible manufacturing system (FMS) is the system where several different products types could be manufactured with a similar level of efficiency for manufacturing mass production of a single product type. An FMS comprises a group of machines with automated material handling equipment's; these machines can be programmed to do some processing operations under the direction, (Smed 2003)[40].

To find a sequence of actions that move the system from its current state to a prespecified goal state, usually refinement strategy is used. This strategy is the most popular one such that it used in most of the classical planning algorithms.

The refinement strategy can be described as follows: A set of candidate sequences is represented to construct a partial plan. This partial plan is used to describe a set of partial solutions by applying planning algorithms on this partial plan, the solutions represented by this partial plan become complete and feasible solutions, (Mathijs 2003)[36].

There are a number of limitations with this classical planning representation. For example, there is no explicit representation of time, there is no provision for specifying resource requirements or consumption and there is no provision for modeling uncertainty.

Over the last few years many extensions of the classical planning problem have been studied: dealing with time (Do 2001; Penberthy 1994; Smith 1999), costs (or utilitymaximization) (Haddawy 1998), limited resources (Wolfman 2001), and planning under uncertainty[17][22][48].

From this view of complexity in planning such systems, we suggest using evolutionary algorithms as optimization techniquesthat are suitable to deal with these extended planning problems because they are considered probabilistic search algorithms and efficient to search a large and complex space of solutions to find a nearly optimal one.

1.4. Multi-Agent Planning

The multi-agent planning problem is the problem where description of the initial state is given with a set of goal states and a set of agents. Each agent defines a set of its capabilities and its private goals. Agent responsibility is to find a plan that achieves its private goals, such that these plans together are coordinated and the global goals are met as well, (Mathijs 2005)[35].

Many tasks require a team of agents to act together in a coordinated way in a complex, uncertain environment and sometimes shared environment. Such tasks involve many agents, and huge numbers of states and possible actions.

It may seem that planning for MAS is similar to doing that for a single agent with just repeating the strategy for every agent participating in solving the problem. In fact, it is not simple as it seems; there are many different arguments that affect the way of planning that we must follow. These autonomous agents have their own sub goals, tasks and priorities; they may also have some privacy. They also may share the environment in which they execute their plans and thus they will affect the state of this environment in uncontrolled manner.

From a deep study to a MA planning problem, this problem could be split to three major smaller problems: a task allocation problem in which we determine for each agent which subtask to perform, an individual planning problem for each of the agents involved this problem study how to ensure that the tasks allocated to the agent can be performed, and a plan coordination problem (how to ensure that the individual planning processes can be integrated into an overall solution).

1.5. Scheduling Problem

"Scheduling can generally be described as allocating a set of resources over a limited time to perform a set of tasks", (Wiers W.C.S. 1997). Scheduling emerges in various domains, such as time tabling scheduling, space missions scheduling, nurse scheduling, aero plane landing scheduling, train scheduling and production scheduling[46].

One view of scheduling, taken by many AI researchers, is that it is a special case of planning where actions are already selected and we are only left with the problem of determining a viable order. Another view takes it as the problem of assigning limited resources to tasks in order to optimize some goals. Scheduling requires reasoning about time and about resources and involves making choices about which resources to use for any given task where several alternative resources that have different costs and/or durations may be available. It may also involve choices between alternative processes, which may have different costs and future implications, as they become available for some steps in a scheduling problem.

Any scheduling problem consists of A finite set of actions or events of a certain duration, a finite set of resources, each having a specified capacity and cost, specification or an estimation of how much each task requires from resource or some of them and a set of ordering constraints on the tasks.

1.6. Integrated Planning and Scheduling

Planning task is defined as finding a sequence of actions that will transfer the initial world into a specified goal state. Naturally, the possible sequences of actions are restricted by constraints describing the limitations of the world. Many methods developed in AI planning, like the STRIPS representation and planning algorithm, are the core of many planning systems. Opposite to planning, scheduling deals with the exact allocation of resources to activities over time that is to find a resource that will process the activity and finding the time of processing. Taking into consideration reserving constraints on products such as deadlines and production precedence constraints, machines constraints such as machine capacity and machine state and many other constraints to achieve a feasible schedule, (Roman Barták(2002))[7].

Planning and scheduling are closely related as the decisions made at the planning level have a strong influence on scheduling. Ideally, the availability of resources would be taken into account on the planning level. The complexity of the overall problem, however, in most cases excludes the possibility of detailed planning over long horizons and thus the two tasks are treated sequentially.

Detecting a performance-based fault or a failure may require a major change to both the plan and the schedule. Measuring performance is necessary to determine corrective actions needed to identify if any changes are required to the plan. The focus should always be on a current plan that ensures a successful achievement of the goals. If the plan, and subsequently the schedule, is not updated continually, it may deviate from its goals.

As mentioned previously, the job of a planner and scheduler, whether manual or automated, is to accept high-level goals and generate a set of low-level activities that satisfy the goals and do not violate any of the agent's operational rules or constraints. Without a reasonable schedule, plans execution may fail or may be delayed. A schedule is the timetable for the successful execution of a plan. The scheduler will determine the amount of time needed and the resources necessary for the accomplishment of the plan's goals.

1.7. Thesis Contributions

This thesis contributes to the research on MAS planning and scheduling in a highly dynamic and uncertain environment in a number of ways:

1-Amulti-agent planning and scheduling problem is discussed for MAS of autonomous intelligent agents in a dynamic manufacturing system and the complexity of this kind of problems is elaborated on.

2-Evolutionary algorithms are defined and proposed as to be the adequate and most appropriate solution to the problem above.

3-Genetic Algorithms are provided as the evolutionary algorithm that chose to be used in solving this problem.

4- Merging the propose methods with GA algorithm to improve the performance it in production system

5-The proposed solution is discussed and explained in the aspects of problem chromosome representation and the genetic operations used with their complete specifications, and finally the fitness function used to determine the most acceptable plan and schedule.

6-The applicability of the proposed algorithm is theoretically discussed, and a clear framework provided for using this algorithm in solving the problem of planning/scheduling of the production system.

7-Experimental results show the flexibility and efficiency of our proposed work.

8- The results of our proposed methods are analyzed in terms of plan and schedule fitness, chromosome size, stability of the shop floor and the time needed to obtain the plan and schedule needed by experimenting with the five level productions structure.

9- Get best utilization for the resources and less idle time as possible as in the production system

10- This system is possible from which to dispense with production engineers in the factory and this is reflected in the cost to the owners of the factories.

11- Solutions resulting from the programmed systems are usually faster and better technical solutions resulting from people in the Production Systems.

1.8. Thesis Overview

In Chapter 2, we will list some of the efforts that spent on solving the problem of planning and scheduling in the manufacturing systems and some other similar systems and some research that related in AI problem and some branch topics of it such as centralized and decentralized multi agent problems in another side we list various of researches that concern in planning and scheduling.

A discussion of the problem theory and the solution adopted by Chen, followed by discussing our proposed enhancement methods. A complete example of the problem is discussed.

In chapter three, we will show how the chen algorithm worked and explained the reason of choosing this algorithm and showed the benefits of using it instead of anotherevolution algorithm.

In chapter four, we will show the results by using old methods and talk about suggestion methods (properties, behaviors) and show the difference in the results before the modification and after it and in addition to that we describe production system and the related environment .

In the last chapters, we will propose some guidelines that help researchers to search about the some solutions of many problems sides in addition to our conclusions from these theses.

Chapter 2

Literature Review

MA planning and scheduling have a great interest by researchers in the last few years. In this chapter some of the researches in the area of MA planning and scheduling are mentioned.

2.1. Multi-agent planning

There are many researchers who put their efforts in finding more efficient and flexible ways to find the most appropriate plan that moves the agent from a current state to some goal state. Listed below some of these researches:

Bartk in his paper (Bartk, R. 2011) gave a survey of possible conceptual models for scheduling problems with some planning features. A comparison is done to find their advantages and disadvantages[11]. Furthermore, analysis of the problems behind industrial planning and scheduling is done after a study within the project whose task is todevelop a generic scheduling engine for complex production environments.

Ferber,j (Ferber ,J. 2005) presents a generic meta-model of multi-agent systems based on organizational concepts such as groups, roles and structures. This model, called AALAADIN, defines a very simple description of coordination and negotiation schemes through multi-agent systems. Aalaadin is a meta-model of artificial organization by which one can build multi-agent systems with different forms of organizations such as market-like and hierarchical organizations. He shows that this meta-model allows for agent heterogeneity in languages, applications and architectures.

He also introduces the concept of organizational reflection which uses the same conceptual model to describe system level tasks such as remote communication and migration of agents. At the end of his paper describes a platform, called MADKIT, based on this model. It relies on a minimal agent kernel with platform-level services implemented as agents, groups and roles.

Edmund H. Durfee and Jeffrey S. Rosenschein in their paper (Edmund H. Durfee and Jeffrey S. Rosenschein,2012) explain the terms that related to work to clarify what they might mean, and encourage the community to consider useful decompositions of the broader research objectives of DAI[11]. For that reason, the reader is forewarned that, in the bulk of the remaining paper, they use of the term "multi agent system" take on the more narrow meaning as was first intended, and as derived from the history of the DAI field .they then consider several views of how multi agent system (MAS) research differs from distributed problem solving (DPS) research. Each of them views provides some insightis important questions in the field, and into different ways of solving problems and designing systems. They conclude by urging the community to not lose track of useful distinctions within the field, and to universally adopt terms to describe distinctive subfields.

Phillip J. Turner, Nicholas R. Jennings (Phillip J. Turner, Nicholas R. Jennings,2006) in Improving the Scalability of Multi-agent Systems paper hypothesize that multi-agent systems need to be bothself-building (able to determine the most appropriate organizational structure for the system by themselves at run-time) and adaptive (able to change this structure as their environment changes).

To evaluate this hypothesis they have implemented such a multi-agent system and have applied it to the domain of automated trading. Preliminary results supporting the first part of this hypothesis are presented: adaption and self-organization do indeed make the system better able to cope with large numbers of agents.

While Timothy J. Norman (Timothy J. Norman, 1997) in his paper (Designing and implementing a multi-agent architecture for business process management) presents a general multi-agent architecture for the management of business processes, and an agent design that has been implemented within such a system. The autonomy of the agents involved in the system is considered paramount. Therefore, for agents to agree on the distribution of problem solving effort within the system they must negotiate.

Mathijs de Weerdt(Mathijs 2003) was interested in finding a way to coordinate planning agents thought revealing all vital information be introducing a formal framework using resources to describe multi-agent plans. His approach has an advantage of enabling the use of a more sophisticated and automated coordination of the plans of organizations[34].

In another publication(Mathijs 2005, Mathijs 2006), he introduced a way to organize current work on multi-agent planning by defining several phases in the multi-agent planning process. And he described some multi-agent planning techniques[35].

Edward F.Watson in (Edward F. Watson 1997) built a simulator for a Make to Order production environment. In this environment, orders are tied to specific customer at the moment of producing them instead of putting them in the stock to be used to service any customer needs them at the moment when they are available.

EithanEphrati (Eithan Ephrati,2002)suggested an approach to multi-agent planning that contains heuristic elements. The method made use of subgoals, and derived sub-plans, to construct a global plan. Agents solve their individual sub-plans, which were then merged into a global plan. The suggested approach reduces overall planning time and derives a plan that approximates the optimal global plan that would have been derived by a central planner, given those original subgoals.

Allen. J (Allen.J, 2000) describeshis experience with combining two interactive agent systems: TRIPS (The Rochester Interactive Planning System) and CAMPS-MP (Constraint-based Airlift Mission Planning Scheduler-Mission Planner), an interactive airlift scheduling tool developed for the US Air Force. This revealed requirements for effective multi-agent mixed-initiative interactions, including the role of explanation and the need for contextual information sharing among the agents.

Watson presented a simulation based resource planning approach that uses simulated lead times (based on queuing in the system) instead of predetermined lead times. His simulator is applied at the macro level to generate order-release plans that are based on realistic shop conditions.

After comparing his simulator with the Material Requirement Planning (MRP) planningapproach in a make-to-order production environment, experiments shows better performance gained by this new simulator compared to the MRP.

Smed and Johnsson (Smed 2003) in their work discussed the problem of production planning in Printed Circuit Board (PCB) assembly[40].

(L. Han 2002) paper describes the mixed-initiative problem-solving features of an

Integrated Process-Planning/Production-Scheduling (IP3S) shell for agile manufacturing.

IP3S is a blackboard -based system that supports the concurrent development and dynamic revision of integrated process-planning and production-scheduling solutions and the maintenance of multiple problem instances and solutions. In addition, it supports flexible user-oriented decision-making capabilities, allowing the user to control the scope of the problem and explore alternate tradeoffs ("what-if " scenarios) interactively. The system is scheduled for initial deployment and evaluation in a large and highly dynamic machine shop at Raytheon's Andover manufacturing facility[27].

Poeck (K. Poeck2012) proposed a support system that covers the whole range from completely interactive scheduling and rescheduling to totally automatic plan generation[31].

In paper (Patig, S. 2001) a planning methodology is adopted. This methodology based on planning steps which can be used for material requirements planning and scheduling[38].

The aim is to cope with uncertainty in production planning.

In FormationConstrainedMulti-Agent Control paper ,Egerstedt. M (Egerstedt. M,

2001) proposea model independent coordinationstrategy for multi-agent

formation control. The main theorem states that under a bounded tracking error assumption, the method stabilizes the formation error. He illustratedthe usefulness of the method by applying it to rigid body constrained motions.

RazNissim (RazNissim, 2010) presents a fully distributed multi-agent planning algorithm. His methodology uses distributed constraint satisfaction to coordinate between agents, and local planning to ensure the consistency of these coordination points. To solve the distributed CSP efficiently, he modified existing methods to take advantage of the structure of the underlying planning problem, multi-agent planning domains with limited agent interaction; his algorithm empirically shows scalability beyond state of the art centralized solvers. This work also provides a novel, real-world setting for testing and evaluating distributed constraint satisfaction algorithms in structured domains and illustrates how existing techniques can be altered to address such structure.

In grand challenge for multi-agent systems paper Kitano. H (Kitano, H, 2000) present detailed analysis on the task domain and elucidate characteristics necessary for multi-agent systems for this domain.

Piotr J. Gmytrasiewicz andPrashantDoshi (Piotr J. Gmytrasiewicz , Prashant Doshi,2005) developed a framework for sequential rationality of autonomous agents interacting with other agents within a common, and possibly uncertain, environment. They use the normative paradigm of decision-theoretic planning under uncertainty formalized as partially observable Markov decision processes (POMDPs) as a point of departure. Solutions of POMDPs are mappings from anagent's beliefs to actions. The drawback of POMDPs when it comes to environments populated byother agents is that other agents' actions have to be represented implicitly as environmental noisewithin the, usually static, transition model. Thus, an agent's beliefs about another agent are not partof solutions to POMDPs.

Stephen Chenney and OkanArikan (Stephen Chenney and Okan Arikan,2006) introduce an efficient algorithm that creates path plans for objects that move between user defined goal points and avoids collisions. In addition, the system allows "culling" of some of the computation for invisible agents: agents are accurately simulated only if they are visible to the user while the invisible objects are approximated probabilistically. The approximations ensure that the agent's behaviors match those that would occur had they been fully simulated, and result in significant speed up over running the accurate simulation for all agents.

While Gerard Gaalman presents the state-of-the-art literature review of the combined MTO–MTS production situations, he proposes a comprehensive hierarchical planning framework that covers the important production management decisions to serve as a starting point for evaluation and further research on the planning system for MTO–MTS situations[20].

Anderson in his paper suggested theory which assumes that serial lists are represented as hierarchical structures consisting of groups and items within groups. Declarative knowledge units encode the position of items and of groups within larger groups. Production rules use this positional information to organize the serial recall of a list of items. In ACT-R, memory access depends on a limited-capacity activation process, and errors can occur in the contents of recall because of a partial matching process. These limitations conspire in a number of ways to produce the limitations in immediate memory span. As the span increases, activation must be divided among more elements, activation decays more with longer recall times, and there are more opportunities for positional and acoustic confusions. The theory is shown to be capable of predicting both latency and error patterns in serial recall. It addresses effects of serial position, list length, delay, word length, positional confusion, acoustic confusion, and articulator suppression[4].

2.2. Multi-agent Scheduling

Cheng (Cheng et al Dec. 2011) in his study improved the results of the classical Economic Lot Scheduling Problem (ELSP). He added a decision variable to the decision variables already defined in ELSP, which is the production rate. He assumed having a single facility, and multiple products are to be produced on this facility and just one product can be produced by this facility at any specific time slot. Another assumption is that setup is required when the production switches from one type of product to another[14].

Both setup times and setup costs are considered. Finally, a constant production rate is assumed. . A cyclic rotation schedule for multiple products is obtained taking into account all previous assumptions. The objectives are to determine the setup schedule and production rate for each product that minimizes the average total costs, which include the inventory, backlog and setup costs.

Wiers in (Wiers, V.C.S., and T.W. van der Schaaf, T.W. 2012) addressed the problem of allocation of tasks between scheduling systems and human schedulers for various types of production units[46].

Ottjes and Veeke (J.A. Ottjes April 2000) presented a simulation approach for planning and scheduling a flow of complex jobs for job shop like production systems.

Production order assumed to be a set of production tasks. These tasks are represented by a direct activity network, each activity in this activity network represents a single production task to be processed on a specific machine (processing machine or an assembly machine)[29]

.Each machine has specific properties which impose some restrictions on this machine, such as relative production speed and setup times and scheduling rules. The task duration may be stochastic having any probability distribution.

Kevin and Gue (Kevin R. GUE 2010) introduced the notion of “almost continuous time” to obtain good solutions to large problem efficiency that solves the model of multiple processor flowshop that results in a computationally intractable formulation, then casting the problem in production planning term and finally, extracting the production schedule from the solution[26].

Edwin (Edwin A. Kjeldgaard) described the implementation of a computerized model to support production planning in a complex manufacturing system. The model integrates two different production processes (nuclear weapon disposal and stockpile evaluation) that use common facilities and personnel at the Pentax plant at the USDepartment of Energy facility. The two production processes are characteristic of flow shop and job-shop operations[19].

In (Roman Barták June 2000) Bartak presented a model implementation that covers most of the industrial scheduling problems. This slot representation for scheduling problems requires some planning capabilities. The main disadvantage of this model implementation is the big memory consumption[8].

2.3. Planning and Scheduling With Evolutionary Algorithms

M K LIM and Z ZHANG (M. K. Lim 2002) introduced a flexible production system copes with dynamicity of the market by introducing a multi-agent system that integrates process planning and production scheduling. This system consists of various autonomous agents that have the capability of communicating with each other and making decisions based on its knowledge[37]. The process of job assignment to machines and the process of handling the negotiation between the different autonomous agents are handled by an iterative bidding mechanism. This mechanism enables optimum process plans and production schedules to be produced concurrently. To deal with the optimization problem (i.e. to what degree and how the currency values are adjusted in each iteration) a genetic algorithm (GA) approach is developed. A test case is used, and the results showed that currency adjustment at a bidding iteration will gradually minimize the total production cost.

DiptiSrinivasan (DiptiSrinivasan2011) presented an integrated framework for generating optimum unit commitment and dispatch schedules. He employs a hybrid technique by which a genetic population can be confined to a set of feasible solutions.

Constraint violation by each member of the population is avoided for both linear and nonlinear constraints by using a heuristics approach. By combining the advantages of

Knowledge-based methods with the strengths of evolutionary algorithms, a reduction in computing time resulted. This reduced computing time makes it possible to use this application in daily operation scheduling[16].

Gonçalves (J. F. Gonçalves 2005) found a solution to the problem of the Job Shop

Scheduling Problem that produces optimal or near optimal solutions on all instances tested from the literature. His solution based on applying a hybrid genetic algorithm, the chromosome representation of the problem is based on random keys. The schedules are constructed using a priority rule in which the priorities are defined by the geneticalgorithm. Schedules are constructed using a procedure that generates parameterized active schedules. After a schedule is obtained a local search heuristic is applied to improve the solution[25].

The computation results validate the effectiveness of the proposed algorithm. The approach is tested on a set of 43 standard instances taken from the literature and compared with 12 other approaches. The algorithm produced solutions with an average relative deviation about 0.39% to the best known solution.

In David Chary paper (Charypar, D. 2005), a genetic algorithm (GA) was presented that constructs all-day activity plans. It uses as input a set of possible activities, and a utility function to score activity schedules. The algorithm is run on several examples, it is shown that the algorithm generates plausible solutions both for crowded and for relaxed activity sets, and that it can do so even when the computation time is restricted. The most important aspect of this work is that arbitrary utility functions can be used[12].

Andy Auyeungn (A. Auyeung 2003) attempted to solve the problem of multiprocessor scheduling. Four common heuristics used by List Scheduling are presented and compared with the proposed multi-heuristic based solution from the view of performance.

List scheduling employs heuristics to choose among all tasks that are ready to be executed. It does this by keeping a list of "ready" tasks which is prioritized based on a particular heuristic. Andy proposed a genetic algorithm that finds a good combination of four common list heuristics to produce a schedule with shortest execution time. The results of the experiments show that scheduling found with the proposed multi-heuristic

List scheduling genetic algorithm outperforms those found with each one of the four list scheduling heuristics alone and for large number of tasks.

Edmund Burke (E.K. Burke 2012) presented a Genetic Algorithm Based University Timetabling scheduling System[11].

Lang and Ross (H-L Fang 2010) improved the previously best known results produced by tabu search on some benchmarks Open-Shop Scheduling Problems (OSSPs). A hybrid Genetic Algorithm is used tomake the system more flexible and easy to use in terms of development time[23].

Andrew (Tuson, A L 2011) presented an implementation of a genetic algorithm to solve the problem of scheduling a production system where there is a number of Products, Machines, Processing times and the maximum time needed by all machines to finish the requested orders. Two performance enhancements, hybridization with a local search algorithm, and a "string fridge" are evaluated [44].

S.Stoppler in his paper (Stöppler, S 1995) took a special case of the general scheduling problem which assumes the existence of N-jobs and M-machines with

Prespecified processing times. He investigates the application of a parallel genetic algorithm (PGA) to this case of the problem.

Harding and N.J in their research aimed to maximize the total net present value in the problem of productions scheduling of a group of linked oil and gas fields. A stochastic search technique is applied using the genetic algorithm. He updated the crossover operator used in the genetic algorithm to become suitable to this specific problem (T. J.Harding 1996).

Bierwirth and Attfeld (Cheng et al Dec. 1998) presented a general model static, dynamic and non-deterministic production environments using Genetic Algorithm. This algorithm is tested in a dynamic environment under different workload situations[9].

Thereby, a highly efficient decoding procedure is proposed which strongly improves the quality of schedules. It is shown by experiment that conventional methods of production control are clearly outperformed at reasonable runtime costs.

Kurbel (K. Kurbel2009) employed a hybrid of the Parallel Recombinative Simulated.

Annealing (PRSA) with the familiar simulated annealing algorithm in order to improve the methods of assigning jobs to the machines in a production system. He assumed that these products must be produced in a pre-specified order[30].

Braune and Wagner (BRAUNE R., WAGNER S 2004) presented an optimization approach to a production planning and control system of a company which produces special purpose vehicles and equipment, he has developed architecture of an optimization system for production planning and scheduling in the manufacturing line of this company[10].

Almeida (Almeida, M. R 2001) in his work developed a method that proved to have an excellent performance to non-provided demand objective and production that can't be allocated in the tanks objective. He used the Genetic Algorithms to solve this scheduling problem and combines it with a rule based system[2].

Chapter 3

Problem description and analysis

3.1. Production Problem Analysis

Production problem can be described from different direction: Agent population, problem domain, resources and constraints. A brief description of each one will be discussed in the next subsections.

3.1.1. Agent Population

Agent population can be described by many characteristics (Mathijs

2005). Agents Quantity, which is the answer of the question "How many agents are employed to solve the problem?" Agent Heterogeneity is how much agents are closed in their characteristics and in the way they use in solving the problem.

And agent's Complexity that refers to how much it is hard to predict what an agent will do.

3.1.2. Problem Domain

The production system domain is highly dynamic. Environment dynamicity appears through the continual state variables in this environment, such as the time, continual orders arrival and machines state change.

3.1.3. Resources

Agent may need to use different resources to accomplish its tasks.

Scheduling shared resources among multiple agents is one of the most difficult responsibilities of a planner and scheduler system. Some of the resources may be affected by the dynamicity of the environment.

3.1.4. Constraints

Constraints are restrictions on some parameters that will affect performing operations by the agent. For example, activities will be affected by constraints on the time, quantity, cost or other constraints.

A successful plan allows the orders to be delivered before its deadline date and takes into consideration the temporal order of sub-products production. Such that some sub-products depend in their processing on the production of other sub-products.

During agent's operations there is a high probability of violating any of the constraints, especially with such dynamic and uncertain environment. These

Constraints violations are called conflicts.

3.2. Chen's Algorithm

3.2.1. Chen's Policy

Chen tries to build a framework for a dynamic production system. This framework builds a schedule in advance to accommodate with this system's high dynamicity. Nowadays, large production systems in the market place enter a high competition to handle the mass orders requests from customers, and to deliver these orders on-time.

A production system with set of machines is studied. Orders are assumed to be arrived in continues basis and at random time slots. Each order arrived encapsulates the main product requested by this order, the quantity requested and the delivery dead line. The main is to find efficient plan and schedule to produce these requested orders and delivering them on-time.

An efficient plan and schedule is that the plan or schedule which makes better machine utilization by reducing machines idle times and increasing orders on-time delivery by reducing earliness and tardiness in order delivering as possible.

In this dynamic system, it is required to handle new orders, but it also not efficient to repeat the process of finding a new plan and schedule for each order arrival, instead, Chen suggested a periodic reschedule where schedule is done at specific time slots periodically.

The suggested algorithm is good enough to handle production dynamicity, but a negative effect is noticed at the shop floor resulted from this periodic reschedule. Instability at the shop floor resulted from the frequent change in products schedules and because of the interruption happened to the machines inhand products processing.

Chen supposed “Frozen Interval” algorithm to handle this instability problem. Frozen interval is a period in which a piece of the old schedule is frozen, such that products in this period will still in their last schedule and no rescheduling needed for them. This strategy minimizes the number of the rescheduled items and thus adds stability to the shop floor.

3.2.2 Chen's Assumptions

Some assumptions are made in the make-to-order manufacturing system:

• There are multiple eligible machines with varying ready times.

• A machine can perform one operation at a time.

• A machine can only works for eight hours a day.

• Each operation can be processed on at most one machine at a given time.

• Operations are non-preemptive.

• Setup times are negligible or are included in the processing times.

• New orders are continuously introduced into the production system on the infinite time horizon.

• MTO (Make To Order) production system is assumed

• Reschedule interval = 8 hours (1 working day).

• Frozen interval = { 2, 4, 6}

• Number of machines : { 4…. 10} Normal Distribution

• Machines ready times: {0 ….. 4 hours}

• Processing time of components (lower level products): {0.1 …… 0.4 of the hour, step 0.1}

• Processing time of sub-assemblies and the final products: {0.3 …… 0.7}

• Cost of idle times: 50 … 100 step 10

• Cost of earliness: 50 …. 100 step 10

• Cost of tardiness: 5 * earliness penalty

• Number of orders arrives at each reschedule point: 0 … 5

• Due dates of orders: reschedule slot + 1 ….. reschedule slot+10

• Order quantities: 5 …. 30 step 5.

• The 5 level Product structure from (Lee 2002) shown in Figure1 is adopted to test the algorithms.

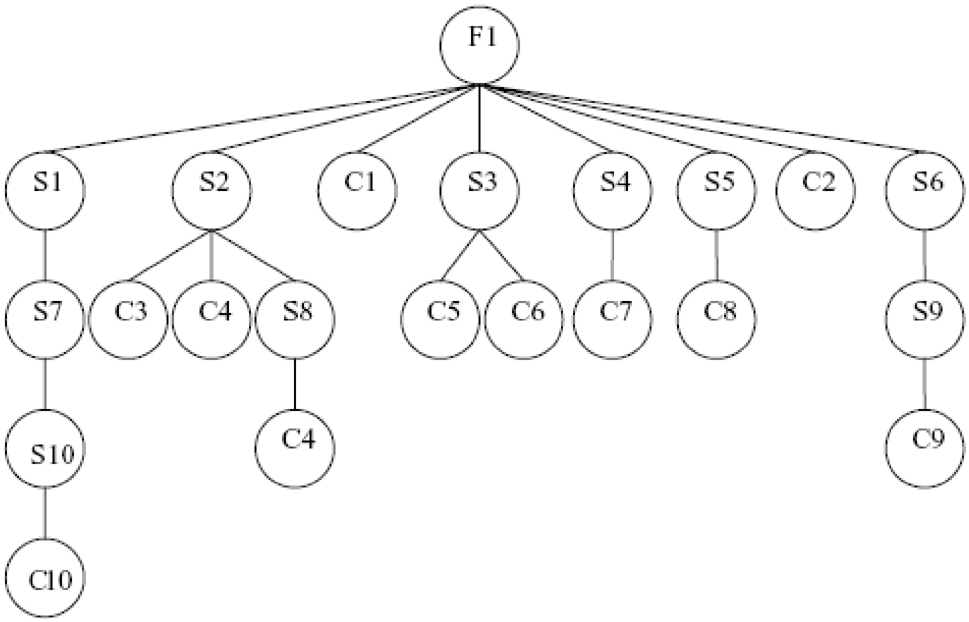


Figure 3.1: The Five-Level Product Structure

3.2.3 Chen's Genetic Algorithm Solution

A genetic algorithm (GA) mimics the evolution and improvement of life through reproduction, when chromosomes contribute with their genetic information to build new chromosomes with better fitness and more surviving chances.

Each 'individual' of the generation represents a feasible solution to the problem, coding distinct algorithms parameters that should be evaluated by a fitness function. GA operators are mutation (the change of a random position of the chromosome) and crossover (the change of slices of chromosome between parents). Ideally, the best individuals are continuously being selected, and crossover and mutation take place. Following few generations, the population converges to the solution that better attend the performance function (James Cunha 2001)[28].

A major advantage of a GA approach is that it is a stochastic-directedsearching technique that does not get stuck at local optimal, but instead looks at the entire range of possible solutions. For complex or highly nonlinear problems (as many real-world problems are), a GA approach is usually the best choice.

3.2.4 Objective Function

In a population of chromosomes, each chromosome has a value; this value represents how much the chromosome is suitable to be adopted as a solution. This value assigned to the chromosome by a problem-specific function called "The Objective Function".

Objective function adds flexibility to the genetic algorithm. Such flexibility allows the use of methods such as look-up tables and if-then statements that allow the function to be discontinuous. So, the objective function represents a metric that should be optimized. An optimization of the objective function, when presented with a solution, assigns to it a numerical value which reflects its quality (T. J. Harding

1996). In the present case, the objective function comprises the costs resulted from any idleness in resources and earliness or tardiness costs of the orders. An individual with a lower fitness represents a better solution to the problem than an individual with a higher fitness value. Those individuals are favored in survival and reproduction,

Thereby shaping the next generation of potential solutions.

Given a chromosome Xh, the fitness function eval(Xh) is defined in Equation1

(chen 2007), which aggregates production idle time, earliness and tardiness penalty:

$$\text{Evalu(Xh)}=\left\{I\left(m.C\max-\sum_{i=1}^{n}\sum_{p=1}^{pi}tipk.Nip.Qi-\sum_{k=1}^{m}rk\right)+\sum_{i=1}^{n}\left[TC*LIi+EC*EIi\right]\right\} \quad (1)$$

Where:

n: number of orders

m: number of machines

pi: final product of order Oi

Qi: quantity of order Oi

Nip: number of items p needed for one final product Pi tipk: processing time required by item p of order Oi on machine Mk (p=1, ..,Pi)

rk : ready time of machine Mk

I: cost of idle time per hour

TC: cost of tardy orders per day per order

EC: cost of early orders per day per order

Cmax: production makespan: the last time slot when all request orders finished

LIi : number of tardy days (integer) for order Oi

EIi : number of early days (integer) for order Oi

3.2.5 Population Diversity:

It is one of the most important factors that determine the performance of the genetic algorithm because it enables the algorithm to search a larger region of space.

Diversity refers to the average distance between individuals in the population, where the population has a high diversity if the average distance is large; otherwise, it has a low diversity.

The parents for a new individual are selected at random out of the current population. When the offspring is better than the currently worst member of the population, then the worst member is replaced by the new offspring. Otherwise, the offspring is not kept. Since there is no selection at the parent level, all existing solutions except the worst are treated equivalently, which maintains a relatively large degree of diversity in the population. (Charypar, D. 2005). In order to maintain diversity in a population, a mutation operator is used (mutation will be discussed later).

3.2.6 Population Size:

Population size reflects the number of individuals in a population. The larger population size helps better optimal solution will be found.

We applied the genetic algorithm on a population of 100 chromosomes in 200 generation. These numbers are chosen after several experiments shown that there is no better solution gained when expanding these values.

3.2.7Chromosome Representation:

It is necessary to come up with a way of representing a solution instance in the computer. This way of representation is referred to as encoding. Before a genetic algorithm can be run, a suitable encoding (or representation) for the problem must be devised. Encoding has a large influence on the potential performance of a GA.

There are various methods to encode the problem, but the most common one is the binary encoding, which consists of fixed bit string (Strings of ones and zeros) to represent certain input data within the problem domain. The following example illustrates a binary encoding for certain chromosome, which consists of three input data for the problem.

Example 1 :Example of chromosome with binary encoding

Chromosome 1: 1100100101010011100110

The first eight bits represent the first data (variable) in the problem and the next eight bits represent the next variable .And so on.

Other encoding techniques are available, such as Real-Valued, Character, Permutation, and Tree encoding. However, the most appropriate encoding is strongly dependent on the problem domain and the environment where the genetic algorithm will operate.

To solve the production planning and scheduling problem, we need to devise a suitable chromosome representation (encoding).Chen in his genetic solution used double genes (has double values from 0 to 1) to construct each chromosome in the population that represents candidate feasible schedules.

We can summarize the genetic strategy as follows:

• Construct a sequence of products from the product tree. This sequence must contain all products needed to deliver each of the orders already requested. Each product must be specified to each specific order (Chen build his system as a MTO manufacture).

For example: if orders O1 and O2 arrived. O1 needs 5 pieces of S7 and O2 needs 3 pieces of S3 then the following sequence will be derived from the product structure used in the assumptions:

O1S7 O1S10 O1C10 O2S3 O2C5 O2C6

Notes: If different orders need quantities of the same product, then we must distinguish these product quantities from each other by concatenating the name of the product with the name of the order which requests it. Another thing to note is that each order and its associated product encapsulates information about its arrival time, quantity, processing time, machine number and delivery dead line.

• Construct 100 random chromosomes of double genes (the number of genes in each chromosome must equal the number of products in the sequence constructed at the first step).

Example: two chromosomes in the population may be:

0.10 0.17 0.23 0.55 0.76 0.94

0.55 0.10 0.94 0.23 0.76 0.17

• Decoding: decoding process must be done to associate every gene (which represents a special product) in every chromosome in the population to the machine that produces this kind of product. This association (scheduling) process must result in a determination of each product start and end time of execution. A timeline for each machine also will exist after this decoding step.

• For each chromosome, the fitness value must be computed.

• Apply reproduction operation: elitist reproduction, by selecting the best 10 chromosomes according to their fitness and put them in the new generation.

• Select two chromosomes using the Roulette wheel selection operation, and then apply the parameterized crossover operator with probability equals 0.75 to produce two new children. Now, select the better two chromosomes of parents and add them to the new generation. This crossover operation will be repeated until 80 chromosomes added to the new generation.

• Until now, the new population has 90 chromosomes. The rest chromosomes will be generated using immigration in which 10 new chromosomes will be randomly generated and inserted into the new population.

Now, we have a new complete population of 100 chromosomes. These chromosomes will have better or nearly the same fitness average as those in the previous generation.

• This genetic procedure will be repeated for each generation from step 4 until 200 generations are generated.

A complete example will be shown in the last section of the theory.

The main advantage of this random keys encoding scheme is that it is easy to attain feasible solutions after executing basic genetic operations. Since the genetic operations are conducted on the chromosomes (the random numbers) the offspring represented by random key vectors can always be interpreted as feasible production sequences. Alogrithm1 below summarize this discussed procedure.

Algorithm 1:

Chromosome Encoding and Decoding for each orderconstruct the sequence of needed products end for each orderdefine a chromosome of length equals to the number of products in the sequence

// start encoding construct the chromosome from double genesfor each geneassociate it with its corresponding product in the sequence end for each gene

Sort genes according to their double values in ascending order

// start chromosome decoding

For each sorted gene

Check if it is ready for processing

If yes:

Compute its start processing time

Assign it to the required machine

Compute its end processing time

End for each sorted gene

3.2.8 Selection:

Pairs of chromosomes are selected from the population to be parents for crossover operation based on their fitness values. Fittest chromosomes are pooled out to produce fittest offspring.

If the selection is strongly dependent on highly fitting chromosomes, then this can reduce the diversity in the population and can result in premature convergence.

Fitness Proportional selection (Routlette Wheel Selection) (Goldberg,1989) is used, in this type of selection each chromosome has a probability to be selected, chromosome selection depends on it's fitness.

This probability will increase for chromosomes with higher fitness. For example, if we have a population of chromosomes with various fitness values, and we sort the chromosomes according totheir fitness then classify them to 4 different classes (ClassA, ClassB, ClassC,ClassD), each class contains chromosomes having fitness values in specific range.

Now, if we want to select chromosome to do crossover operation on it, we will look at its' fitness. If this fitness is in ClassA then the chromosome has 50% probability to be selected for crossover, but if it was in ClassB, then it will has probability of 30%, and

15% if it was in ClassC. Otherwise, it will have only 5% probability to be selected.

These probabilities appear clearly in Figure 3.2. If we imagine this figure as a wheel and we want to roll this wheel, then for sure, chromosomes with higher fitness (ClassAchromosomes) will have a better chance to be selected.

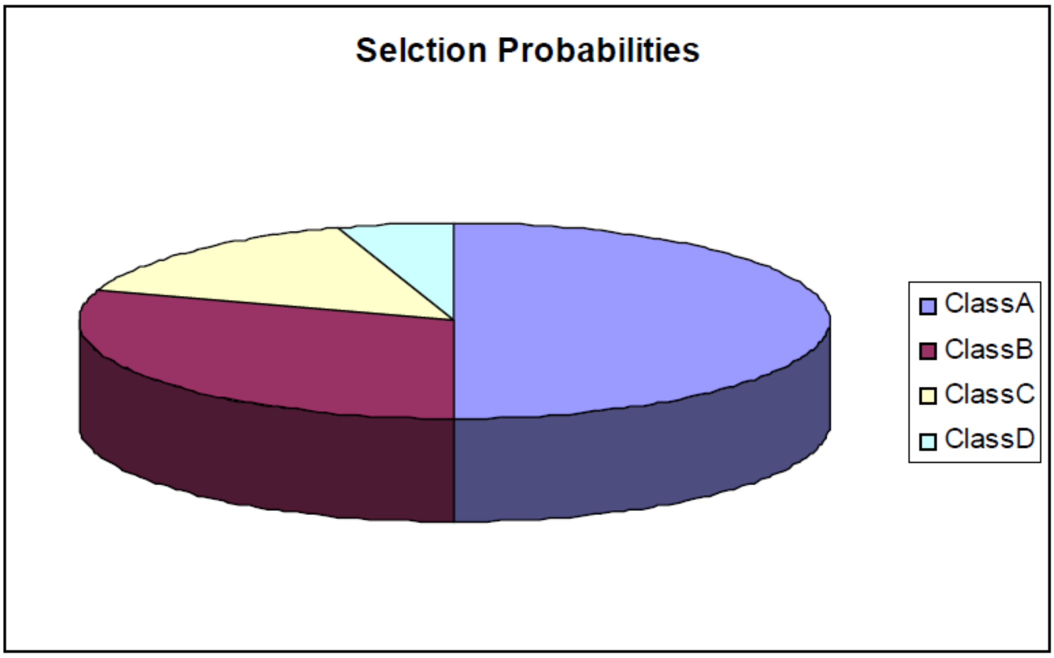


Figure 3.2: Roulette Wheel Selection

There are other types of selection operators such as Sigma Scaling selection and

Rank Selection operators. But the Roulette Wheel selection is the most popular one.

3.2.9 Crossover Operator:

It is a procedure in which a highly fitting chromosome is given an opportunity to reproduce by exchanging pieces of its genetic information with other highly fitting chromosomes.

There are many ways to perform crossover:

• Single-Point Crossover: It is the most common form of crossover operation. In this type of crossover, a single point is chosen randomly fromthe two parent chromosomes. Then, one of the two parts around the selected point is exchanged between these two parents resulting in two new chromosomes. Example3 illustrates the idea:

Eaxmaple3 : Example on single-point crossover function

Parent 1 : 1000110101100111100 11000

Parent 2 : 0011001100111100010 10101

Crossover point : 11

After Crossover, the new offspring are:

0011001100100111100 11000

1000110101111100010 10101

Single-point crossover is not efficient due to the limitation in the number of ways the chromosomes can be split and joined. That will produce “Position

Bias” problem in which the position of input data in the chromosomes will affect their ability to be combined with other input data. Therefore, it is difficult to produce certain combinations of input data, and it may take several generations to generate certain combination.

• Two-Point Crossover: two crossover points are selected randomly, where from the beginning of chromosomes to the first crossover point is copied from the first parent, The part from the first to the second crossover points is copied from the other parent and the rest is copied from the first parent again. This method used to overcome "PositionBias" problem of the single point crossover since it allows for more possible combinations of the chromosomes during crossover operation, but can't produce all possible combinations and can be more likely to cause disruption between related input data.

• Parameterized Uniform Crossover: a random binary vector is created to let crossover occur at any point in the chromosomes. Where if the value in the vector is one, the corresponding input data is copied from the first parent; otherwise, the input data is copied from the second parent. This allows for the greatest number of possible outcomes from the crossover, but can also be disruptive to related input data in the chromosomes.

Parameterized uniform Crossover is applied. This parameterized uniform crossover operation has shown to be computationally better than the one-point or two-point crossover (Hadj-Alouane&Bean,1997). Parameterized crossover operations described clearly in the Example at the end of the chapter.

In Chen's system, 80% of the new population is generated using the crossover operator with probability equals 0.7.

3.2.10 Mutation Operator:

A mutation operator inserts random modules to maintain diversity.

Immigration is used, which is a kind of mutation which involves randomly generating one or more entirely new chromosomes and inserting these new members into the population. In this way, immigration maintains diversity and prevents premature convergence of the population (Bean, 1994; Hadj-Alouane&Bean, 1997). Mutation is applied with probability of 0.95 to generate 10% of the next generation population.

3.2.11 Reproduction:

The best individual chromosomes are directly copied from one generation to the next; this also called “Elitist Reproduction”. In Chen’s system, reproduction is applied to 10% of the current population.

3.2.12 Stopping Criteria:

The simulation is done for 120 hour (15 working day, for 8-hour working day) and is repeated 100 chromosomes to obtain accurate results as possible. At each run genetic algorithm will be called at each replan and reschedule point (if there is products need schedule). In genetic algorithm 200 generation of 100 chromosomes, each one is constructed to get best fitness (best schedule) in reasonable time. Numbers 100 and 200 are used by Chen, and in our study we also try different numbers, but those used are the most suitable ones, since fitness values reach a stable state after 200 generation of 100 chromosome. And the number of chromosomes could not be increased more because this will reflect the time needed to get the schedule negatively, which is not logical in highly dynamic environment.

3.2.13 The Genetic Algorithm Procedure Summary

The main steps in genetic algorithms are shown in Figure 3.3:

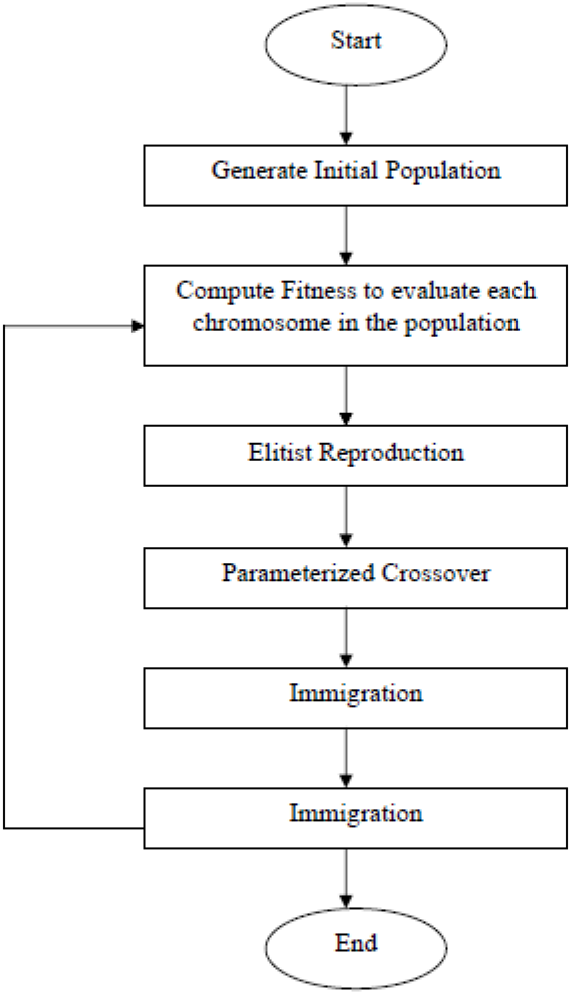


Figure 3.3: The Genetic Algorithm Steps

3.2.14 Drawbacks in Chen's Algorithm:

After careful analysis to the Chen's algorithm, we noticed that:

• Most of the products included in the previous schedule are rescheduled again at the reschedule point due to the fact that precedence relationships between them force some of them to wait until their subassemblies complete processing. The most of waiting timeis not finished till the new reschedule point reached.

• Genetic Algorithm processing time (processing 100 chromosomes of approximately 14 genes each over 200 generation at each reschedule point) is wasted for those tasks that need rescheduling.

• Chromosome size is too large compared with the tasks that exactly get use of their supposed schedule. In GA, smaller chromosome means less processing time and less delay, and thus more dynamic and adaptive system.

• Fixed reschedule interval (8 hours) is not suitable for this dynamic production system where orders arrive at an un-expected time slots andwhere there is no idea about the distribution of orders arrival times over the time horizon.

• Fixed frozen interval is not adequate since it effect and affected by the reschedule interval and the time needed to produce the new schedule.

• The proposed system by Chen is not flexible enough to deal with scheduler and planner failure which makes a disaster in this dynamic environment.

3.3. The Proposed Methods:

From these noticed points, we proposed four enhancements in order to increase the efficiency and flexibility of this production system. To increase efficiency, we implement our proposed methods: “On The Shelf”, “Adaptive Reschedule Interval” and “Adaptive Frozen Interval”. Furthermore, A MAS is adopted to increase this system flexibility.

A block diagram for these classes is shown inFigure 3.4:

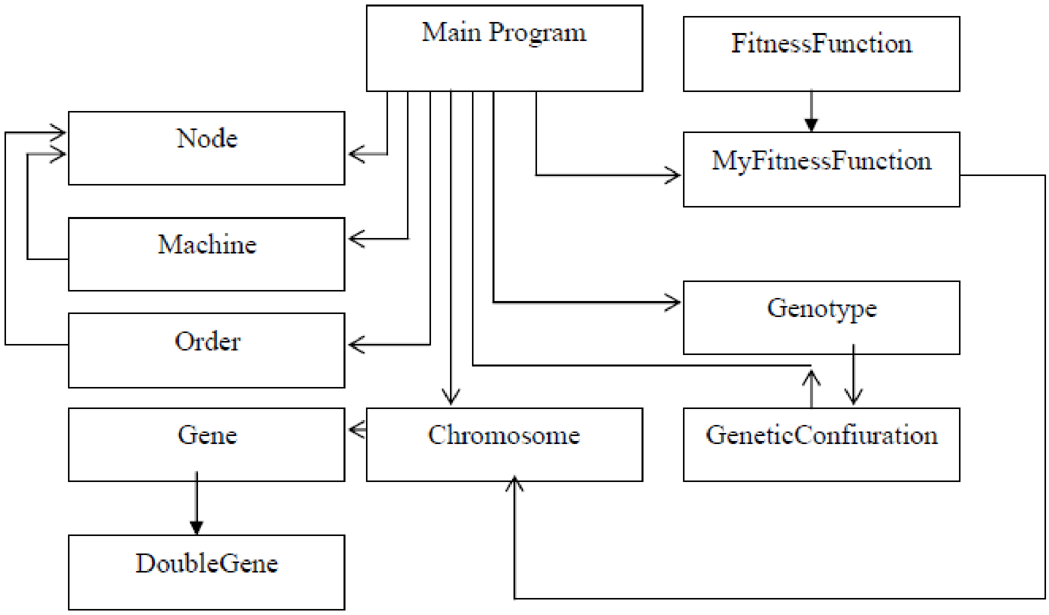


Figure 3.4: Classes Block Diagram

3.3.1 on the Shelf

This idea states that not all needed products will take the chance to schedule them at their schedule or reschedule point. Some of the products have higher priority to enter the genetic-based scheduling process. Our heuristic to select products from the existed sequence in order to give them to the scheduler and planner depends on how much this order is ready to begin execution. Product readiness depends on the location of the product in the product tree and also on the state of its child products (sub-assemblies) in the product tree. According to this heuristic, high level products that have less priority of processing and whose sub-assemblies are not finished yet, will be put on the shelf waiting for the next reschedule point to compete another time for a chance to enter the genetic-based rescheduling procedure.

You may think that some additional negative waiting time will result from this "On The Shelf" strategy, but experiments show that these products were most of the time have normally wait because of the large processing times required to finish their sub-assemblies.

3.3.2 Adaptive Reschedule Interval:

From the fact that the dynamicity of the production system is a basic characteristic of any production system, there is a need to make the proposed algorithm more dynamic, so that it can deal with environment changes in an efficient way. A proposed adaptive reschedule interval is adopted. Any adaptivereschedule interval will be affected by many of the environmental factors such as machines wasted time from the old reschedule interval and the number of tasks needs rescheduling at every reschedule point.

After collecting the factors that affect and affected by the reschedule interval, we conclude Equations2 below, which determines the length of the new reschedule interval depending on how much the previous schedule was efficient and the maximum expected time for any task to be executed.

NRS = CTS + NRI ……………………………………… (2)

Where:

NRS: Next Reschedule Slot

CTS: Current Time Slot

NRI: New Reschedule Interval.

Almanda suggested equation for the estimate nominal scheduling in production system.

ScheduleInMonths = 3.0 x EffortInMonths ^ (1/3)…………………(3)

3.3.3 Adaptive Frozen Interval:

Reschedule interval and the frozen interval are strongly related to each other. Such relationship makes them affect and affected by each other. Of course, the frozen interval must always be smaller than the reschedule interval. In this stage of study we are attempting to find the best percent of the reschedule interval that must be given to the frozen interval in order to obtain a compromised fitness and stability values.

3.4. Performance Measures:

System performance is measured from four directions: fitness, stability, chromosome size and time.

3.4.1 Fitness:

This metric reflects the cost wasted as a result for machine idleness and orders early delivery of delayed delivery. Thus, smaller fitness values are preferred.

3.4.2 Stability:

From the equation of stability below (Equation4) smaller values are preferred.

So, when the frozen interval is very small compared with the reschedule period, most of the tasks existed in the old schedule and not scheduled yet will be rescheduled.

This affects negatively the stability of the shop floor, while when the frozen interval get closer to the reschedule period (here equals 8 hours) most of the tasks in the old schedule will be frozen, and thus no old-new schedule differences found which will lead to better stability and thus we get smaller stability values. But, very large frozen interval will affect negatively thefitness value. This negative effect caused by thedelay cost which resulted from delaying the scheduling of previously arrived orders. In this case, they will not just wait for the reschedule point to be reached. In fact they will also wait the end of the frozen interval.

Equation5 shows the minimum waiting time required for each order to begin processing.

$$\text{Stability} = \quad \sum_i |t_i - t'_i| + \sum_i \text{PF}(t_i - t + t'_i - t) \qquad \text{..........} (4)$$

'

t : current time.

ti: operation starting time in the original schedule.

t′i: the operation starting time in the new schedule.

PF(x) : is the penalty function and it equals 10/x 0.5 , when the totaldeviation from the current time is zero, the penalty is assumed to be zero.

MIN (WT) = NRP – AT + FI …………………..……………… (5)

Where :

WT : order Waiting Time

NRP : Next Reschedule Point

AT : order Arrival Time

FI : Frozen Interval

3.4.3 Chromosome Size:

Computed as the number of genes in the chromosome, chromosome size is one of the main metrics that affects greatly the time needed to find the new schedule using the genetic algorithm. Processing population chromosomes over several generations, and fitness computation for eachnew child, in addition to the time needed by genetic operators (especially which work on the gene level such as parameterized crossover), all of these affected by the chromosome size.

3.4.4 Time:

In dynamic environment where a decision is to be made quickly as possible to adapt with the current state of this environment, "Time" is the main measure for how much your system is adaptive with this highly dynamicity in the environment.

3.5. Complete Example on Chen's Method

Example:

If orders O1 and O2 arrived to a factory of 4 machines and Order1 requests

5 items of S7 with deadline 2dayes (16hours maximum) and O2 requests 2 items of S3 with deadline1 day (8hours maximum).

then a sequence of the requested products will be constructed as follow:

S3O2	C10O1	S7O1	C5O2	S10O1	C6O2

A corresponding chromosome will be constructed of genes of random numbers between 0 and 1. Each gene represents a priority value for executing its corresponding product from the above sequence.

0.23	0.10	0.94	0.55	0.76	0.17

Now, to encode this chromosome to a feasible schedule we will sort these orders according to their gene values. Sorted chromosome is shown below:

0.10	0.17	0.23	0.55	0.76	0.94

Table similar to Table3.1 below must be given for specific manufacture. This table show for each product of the products that can be produced what is the machine that produces this product and what duration does it take to produce one entity of it.

Item	Machine Number	Processing Time (hours)
S3	M1	0.5
S7	M2	0.4
S10	M2	0.3
C5	M4	0.2
C6	M4	0.2
C10	M3	0.1

Table 3.1: machines and processing time required by each product

Assume also that machines required to setup every 8 hours. Setup times assumed for these four machines are in Table 3.2 below.

Machine Number	Setup Time Required (hours)
M1	3
M2	2
M3	2
M4	1

Table 3.2: Setup times for the machines in the example

Step1: the first gene in the sorted sequence is corresponding to the product

C10O1 (assuming that this product will not use any other product to start processing.

In other words, it is a leaf product in the factory product tree in the product tree (see

Figure1)). Beginning form this assumption, an order will be given to the specialized machine to produce the amount needed of C10 for Order1. The start processing time and finish processing are calculated according to equations 6 and 7.

SPT = MAX(MCS,MAX(CEP)) ……………………………….. (6)

Where:

SPT: Start Processing Time for this product

MCS: Machine Current Slot (after ready times and the last processedproduct finish time).

CEP: Child End Processing (for each child product of this product inthe product tree. That is, it is the pre-requested product for thisproduct to be produced).

EPT= SPT + NOI * PT …………………………………….. (7)

Where:

EPT : End Processing Time

SPT : Start Processing Time

NOI: Number Of Items required from this product to deliver the corresponding order.

PT: the Processing Time needed to produce one item of the product.

After this step, C10O1 will be associated to the machine M3. C10 needs 0.1 of the hour to produce a single entity. So, to produce 5 items to O1 we need 0.5 of the hour. Additional 0.1 added as a time for the machine to switch from task to another.

M3 needs 2 hours to setup, thus C10O1 will start at 2 until 2.6.

Step2: the next element in the sorted sequence is C6O2, it is ready for processing because it is a leaf element in the production tree. C6O2 will be associated to the machine M4. C6 needs 0.2 of the hour to produce a single entity. So, to produce

2 items to O2 we need 0.4 of the hour. Additional 0.1 added as a time for the machine to switch from task to another. M4 needs 1 hours to setup, thus C6O2 will start at 1 and finish at 1.5.

Step3: S3O2 is not ready for processing because not all needed sub-assemblies available.

Step4: C5O2 is ready. Will be produced on M4 during the period 1.5 – 2.0

Step 5: S10O1 will be assigned to M3 from 2 to 2.6 to produce 5 items for

order1.

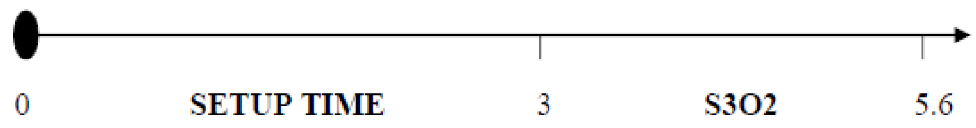
Step6: S7O1 is ready for processing. It will be assigned to machine M2 from the time slot 4.2 until 6.3.Note that M2 spent an idle time between 2 and 2.6 in this schedule. This occur because S7O1 needs C10O1 as a sub-assembly, and this subassembly will be finished at 2.6, so that we could not associate S7O1 to the machine M2 before 2.6 although the machine was available at this time period.

Step6: Now we will return to check the product S3O2, it is now ready forprocessing. Associate it to machine M1 from the time slot 3 until 5.6.

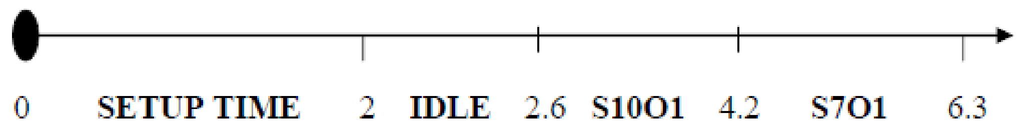
Step7: All products are produced in this schedule, so we obtained a feasible schedule, but how much it is efficient depends on schedule's fitness that will be computed at the next step.

The final machines timelines are shown below.

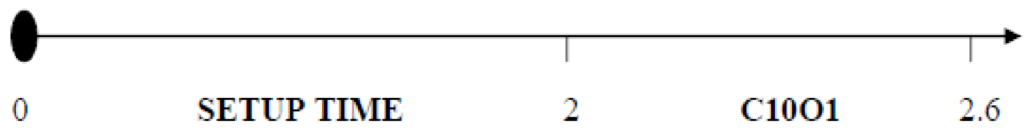
M1 Time Line:



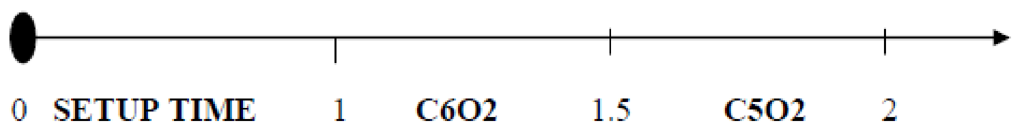
M2 Time Line:



M3 Time Line:



M4 Time Line:



Step8: Apply the fitness equation presented previously to compute the fitness of this schedule.

$$\text{Evalu(Xh)}=\left\{ I\left(m.C\max - \sum_{i=1}^{n}\sum_{p=1}^{pi} tipk.Nip.Qi - \sum_{k=1}^{m} rk \right) + \sum_{i=1}^{n}\left[TC*LIi + EC*EIi\right] \right\} \quad (1)$$

Fitness = 60 (6.3 * 4 – [0.6 + 0.5 + 2.6 + 0.5 + 1.6 + 2.1] – [3 + 2 + 2 + 1]) + 50 *

((int)[(16 – 6.3)/8] + (int)[(8 – 5.6)/8]) + 250 * 0

Fitness = 558 + 50 = 608$

Step9: Repeat this procedure for every chromosome in the population

Step10: Select the best 10 chromosomes (with lower fitness values) and put them in the new generation.

Step11: Use the Routlette Wheel selection to select two chromosomes for crossover operation, and repeat the crossover until producing 80% of the new generation.

We will describe in simple steps what exactly will happen:

If we select these two chromosomes from the population using the roulette wheel selection strategy discussed previously, then crossover can be done as follows:

• Construct an empty chromosome (child chromosome) has the same number of genes such that in the parent chromosomes.

• Chose randomly a value between 0 and 1.

• If the value selected equals 0 then copy the first gene value from the first parent to the first gene in the new child chromosome. Else, copy the first gene value from the second parent to its corresponding position in the child chromosome.

• Repeat the steps from 1 to 3 for every gene in the chromosomes.

0.69 0.37 0.41 0.55 0.98 0.22 0.69 0.37 0.41 0.55 0.98 0.22

If for example, the values tossed are: 100110, then the resulted chromosome is shown below

0.69	0.10	0.94	0.55	0.98	0.17

Step12: Add new randomly generated chromosomes until constructing 10% of the new population.

3.6. Implementation Language

We implement this strategy and the enhancement methods using PHP

Programming Language at windows7 operating system with

1.78 GHz CPU and 256 MB of RAM and XAMPP ,it use to fix related programs such as SQL that be useful for store and import huge data, and apache and we use google chrome explorer because it has proprieties that enable us to make local link between different pages

Chapter 4
Results and Analysis

4. Results and Analysis

The algorithm described in (Chen 2007) is implemented at single agent. 100 runs are done in order to gain accurate results as possible. Additional algorithms are proposed to make enhancement to the original algorithm. Algorithms are implemented in single agent and the main algorithm implemented also on multiple agents production systems.

When applying the Genetic Algorithm, Fitness values will converge gradually to a near optimal value. Below at Figure4.1 you will see fitness values for the 100 chromosomes in the first generation. They are clustered between 3700 and 5500. On the other hand, if you look at Figure 4.2, Fitness values for the same chromosomes after 100 generation are covering the area between 3200 and 5000 which means that fitness values get close to a better one fromgeneration to another.

Note that the amount of dispersion in the figure 4.2 is less than it was in the figure 4.1 and this shows that the stability and consistency is heading for the better, where whenever a large dispersion shows that the way in moving towards random.

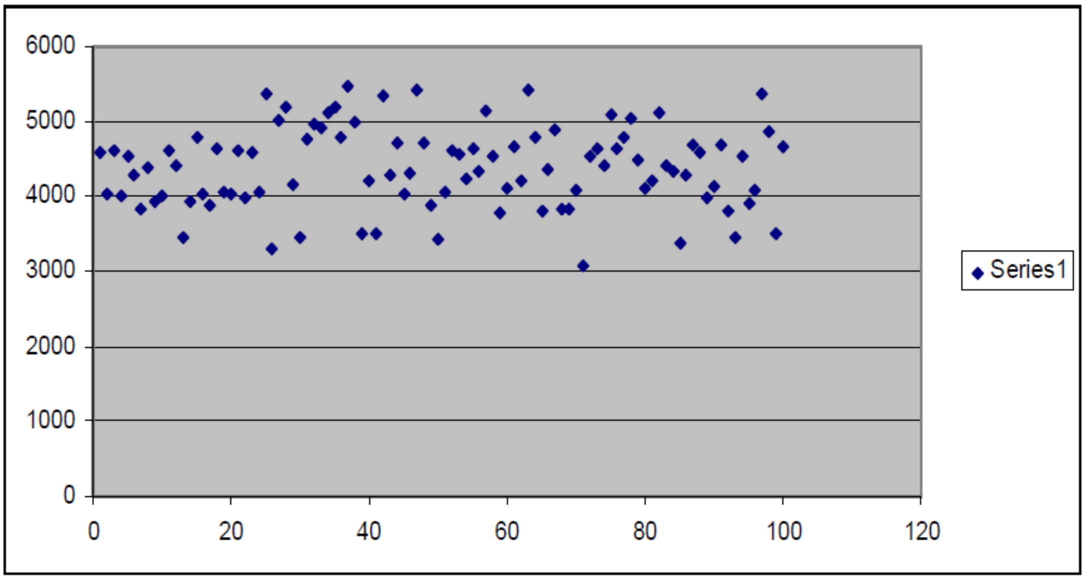


Figure 4.1: Fitness Values of Chromosomes at the first generation

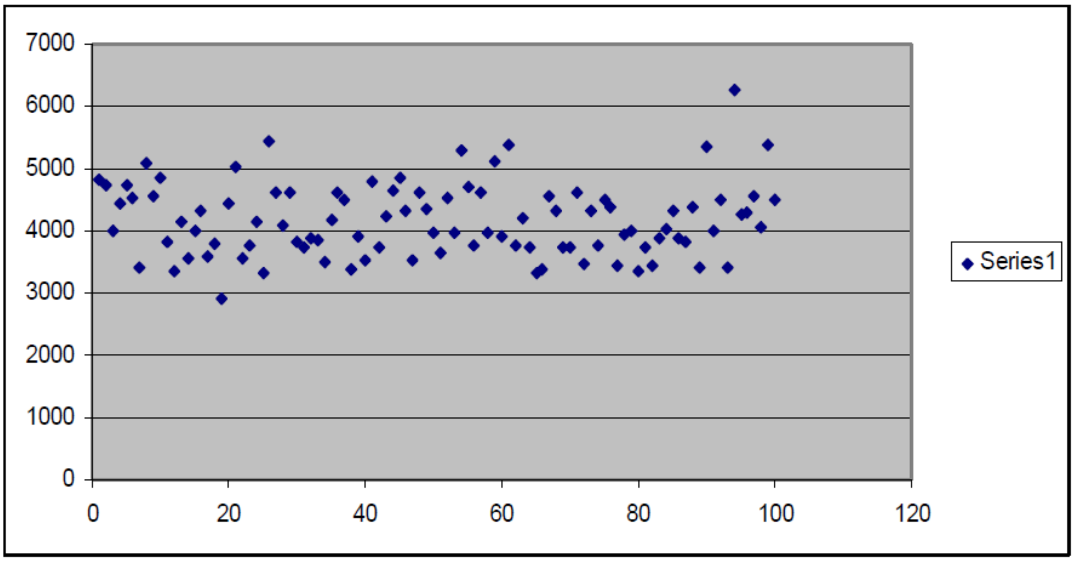


Figure 4.2: Fitness Values of Chromosomes after 100 generation

4.1. Chen's Algorithm Results:

The main algorithm is tested on five orders of products taken from the 5-level order structure shown at Figure 3. The factory has 4 machines with each one has a capacity equals 8 hours, such that it can work 8 hours without re-setup. From this point we make a simulation for 120 hours (15 working days). In these days five different orders arrived on the system with a predetermined deadlines and quantities. Quantities values vary between 5and 30 (step 5).

As mentioned previously, when we talk about the fitness function, the cost of the timethat the machine spent idle without processing any product has a great effect on the fitness of the suggested schedule. We assumed that every hour spent by the machine in idle state costs 60$, and every day the order finished earlier than the desired due date is assumed to be 50$, on the other hand this earliness cost multiplied by 5 reflects the tardiness cost foreach delayed day per order.

Lastly, we assumed that the reschedule point (reschedule time slot) to be at each work day beginning (every 8-hours). All the assumptions taken into account are the same as those in (Chen 2007).

Figure4.3 shows the fitness averages when applying Chen’s algorithms on differentfrozen intervals. From the figure, fitness values are increasing while increasing the frozen interval. This increment in fitness means more cost resulted from machines idle times and/or orders earliness or tardiness which results in a degradation of the performance.

Due to the fact that making a piece of the old schedule frozen means a delay in starting the new schedule, which will reflects the start and end time of processing the products of the new schedule.

Furthermore, this delay in finishing the products in the new schedule will cause tardiness in delivering all orders. In its computation on earliness and tardiness in orders delivery, it will be increased resulting in a slightly worse schedule. On the other point of view, Figure8 shows how stability was affected by increasing the frozen interval.

From the figure, stability values will decrease while increasing the frozen interval, thisdecrement can be explained as an effect to the minimization happened to the number of products that need rescheduling.

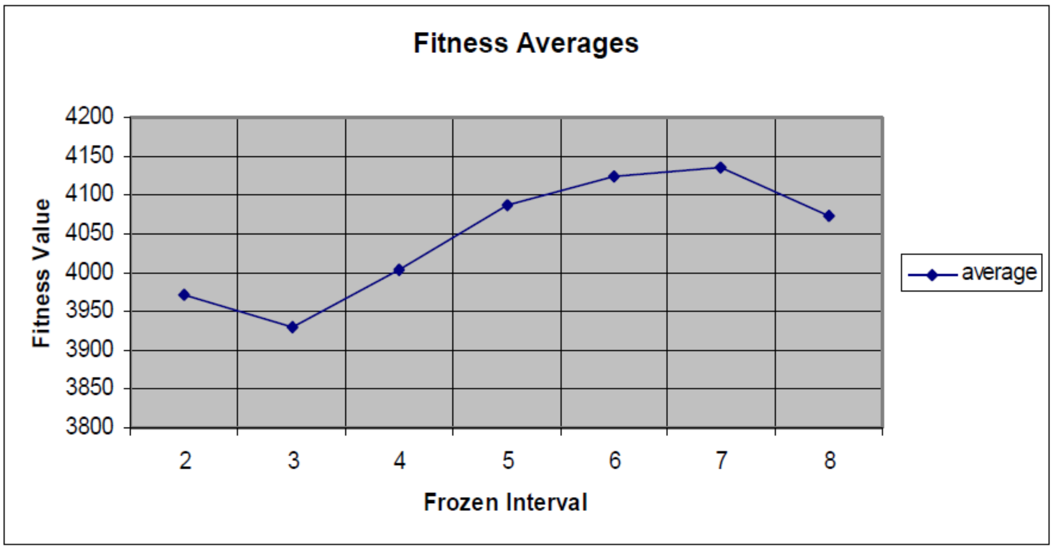


Figure 4.3: The Chen's algorithm Fitness Chart for 100 Runs

Again, increasing the frozen interval means to froze a piece of the post schedule, which led to less products enter the next reschedule, that is fewer products will change their schedule and better shop floor stability gained (lower stability values).

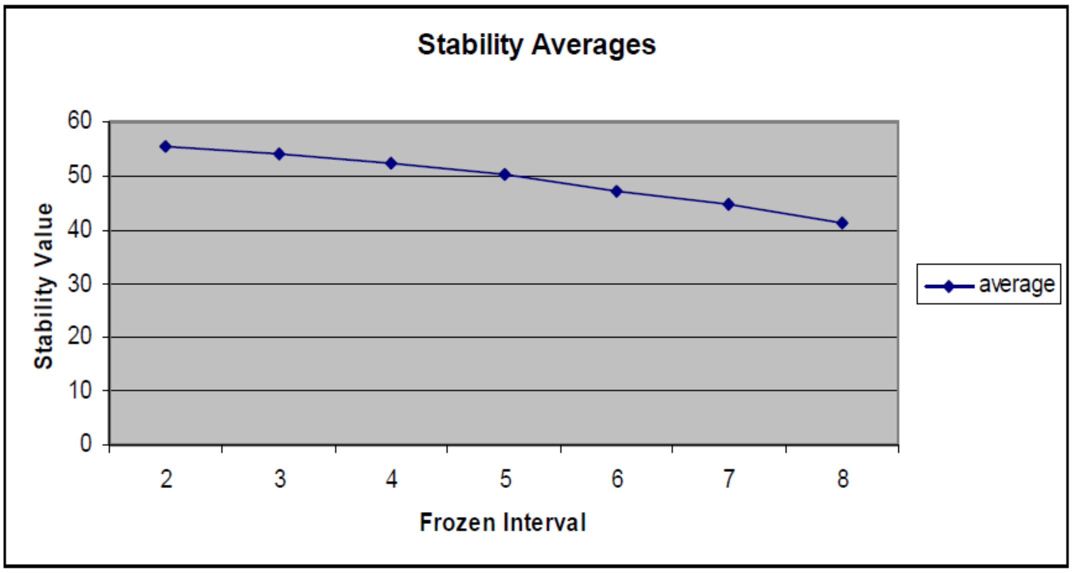


Figure 4.4: Stability values in Chen's algorithms for 100 run with different frozen intervals

Now, we will study the effect of changing the frozen interval on the averagechromosome size. From Figure4.5, chromosome size is decreasing with the increment of the frozen interval. More frozen leads to less chromosome size at the next reschedule point. A deeper look at Figure4.5, we can notice that the average of chromosome size is not decreasing sharply; it is just percent's of a unit (gene number). This explains the stability noticed in the average time needed by the genetic function to find a near-optimal plan and schedule as shown in Figure4.6.

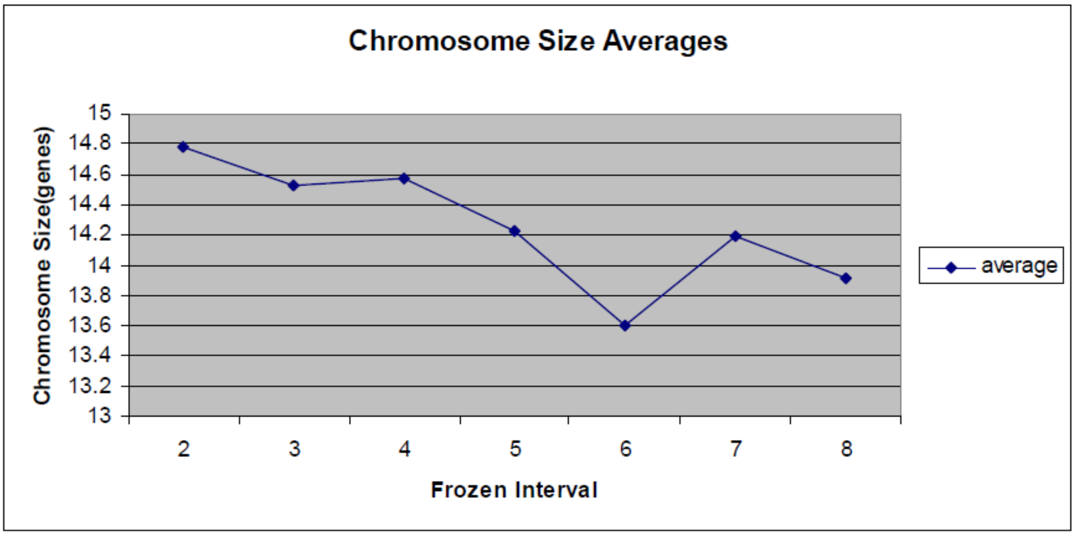


Figure 4.5: Chromosome Size averages for different frozen intervals in Chen's algorithm

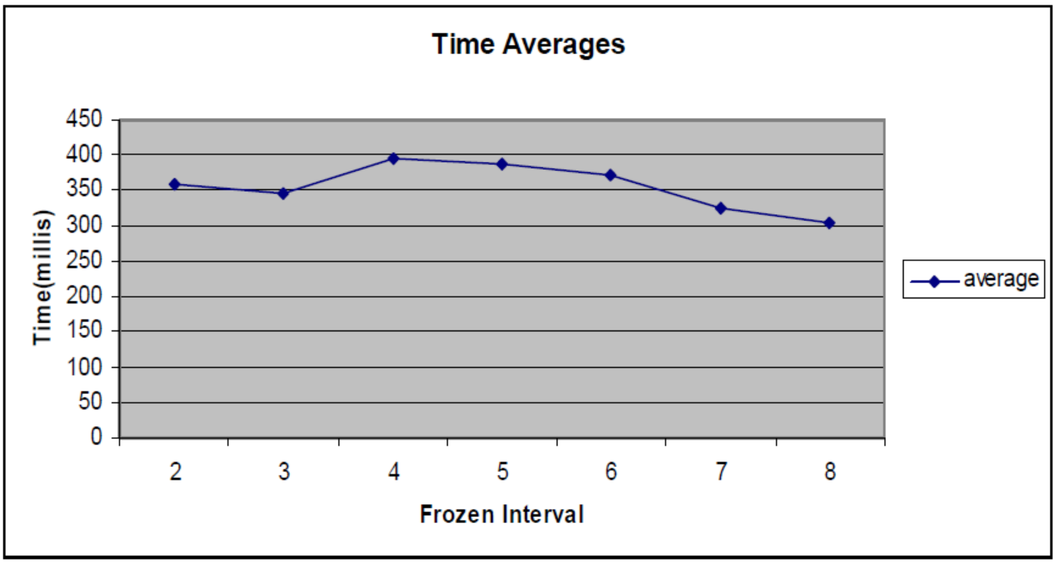


Figure 4.6: Time averages for different frozen intervals in Chen's algorithm

As a summary, the fitness values are goes to the worse while increasing the frozeninterval. On other side, the stability gets better. As discussed before, this is normal because increasing frozen interval will decrease the number of tasks that needs reschedule, this will results in a best stability, while also delaying the new tasks because they will not be schedules until the end of the frozen interval, which of course results in order tardiness and thus increase the cost, the fitness value and worse results gained.

4.2. On the Shelf Idea Results:

Figure4.7, applying "On the Shelf" method results in better fitness values compared to the results obtained when applying Chen's method. These values reflect an enhancement in the fitness values which means better schedule is obtained.

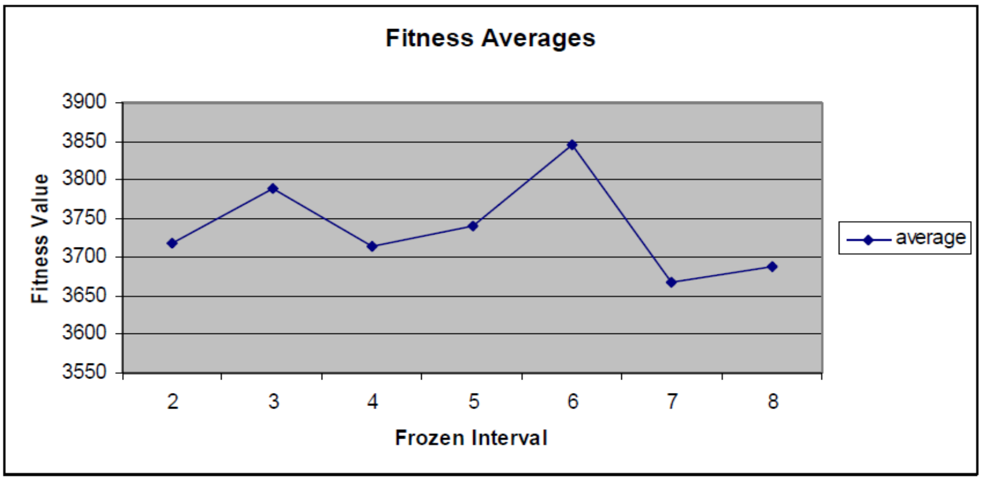


Figure4.7: The Fitness Values for the "On The Shelf" method

If we look at the stability figure (Figure4.8), we notice that stability values rangesbetween 25 and 42, which is a great enhancement compared with those from Chen's algorithm. This enhancement is due to the fact that fewer tasks needed to be rescheduled.

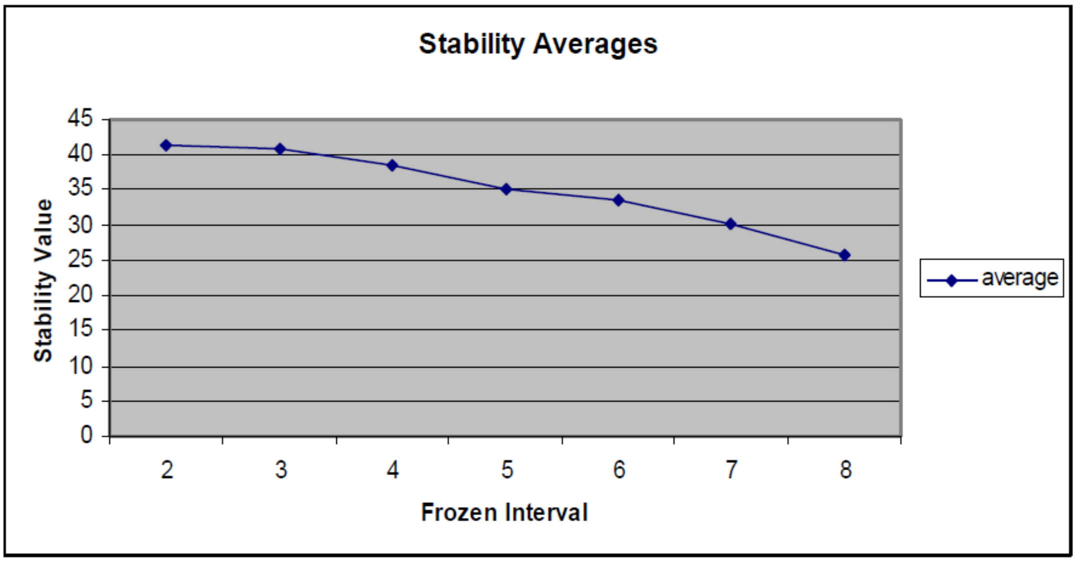


Figure4.8: Stability Values for the “On the Shelf” idea

When looking at Figure 4.9, you will notice how much the new method influence the time required by the genetic algorithm, which is one of the significant factor in any dynamic system.

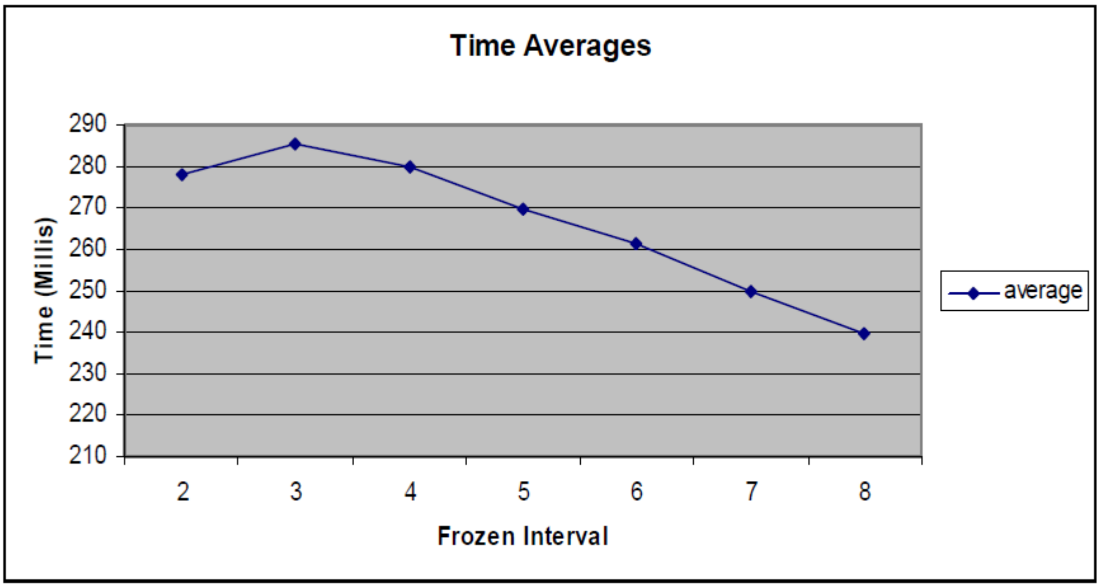


Figure 4.9: Average of time for different frozen intervals collected from “On The Shelf” method

Figure 4.10 shows that the average of chromosome size decreased from the averagestored in the previous algorithm. Also we can notice that chromosome size participate in an opposite relation with the frozen interval; such that when frozen interval increased, the chromosome becomes smaller.

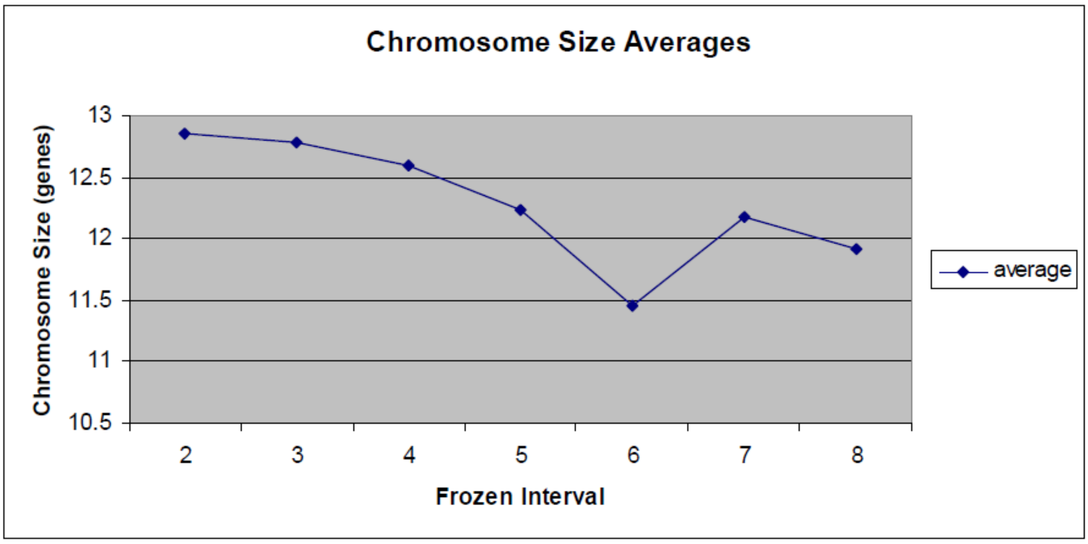


Figure 4.10: Average of Chromosome Size for different frozen intervals collected

from “On The Shelf” method

An enhancement is noticed in all aspects, which is a great enhancement (see Table4.1).

These enhancements percent's are calculated by Equation8 below:

Enhancement = (Chen’s result – new result)/ Chen’s result * 100%.

Metric \ Frozen	2	3	4	5	6	7	8
Fitness	0.063935	0.036082	0.07189	0.084691	0.084282	0.113518	0.094372
Stability	0.256341	0.246866	0.266323	0.299561	0.290493	0.323015	0.375212
Time	0.224041	0.193847	0.291961	0.305668	0.298507	0.232322	0.211635
Chromosome Size	0.130582	0.119752	0.135209	0.139845	0.123853	0.137394	0.143063

Table 4.1: Enhancement Percentage obtained by “On The Shelf” strategy

4.3. Adaptive Reschedule Interval:

Better fitness values are collected by applying this methodology. As shown in

Figure 4.11 below, Fitness values are better than the results shown in Figure4.3 (results from Chen).

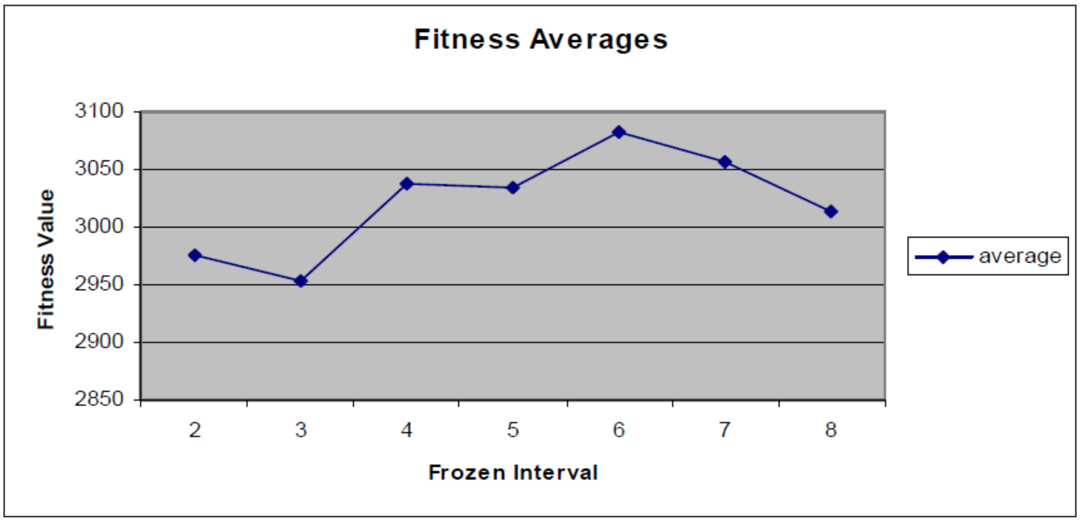


Figure 4.11: The Fitness Values for “Adaptive Reschedule Interval” method

Figure 4.12, shows the stability values. When comparing these values with those by

Chen, better values are clear. Thus, adaptive reschedule interval increased the stability of the shop floor while on the same time enhancing fitness values.

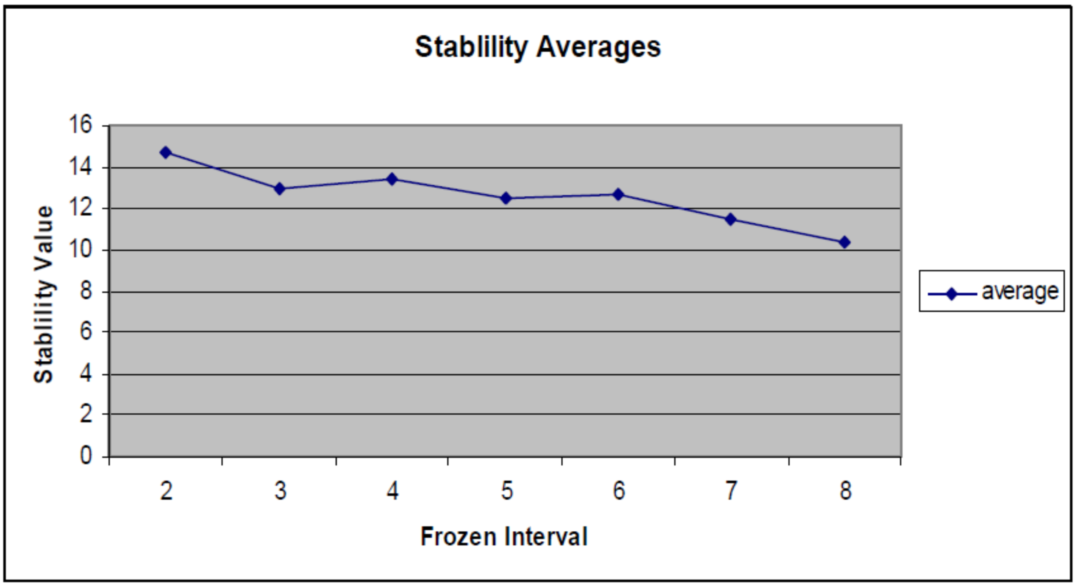


Figure4.12: The Stability Averages for “Adaptive Reschedule” method

Time averages are show at Figure 4.13. The same behavior of the time curve innoticed as that in the “On the Shelf” curve, lesstime means more dynamicity less time wasting.

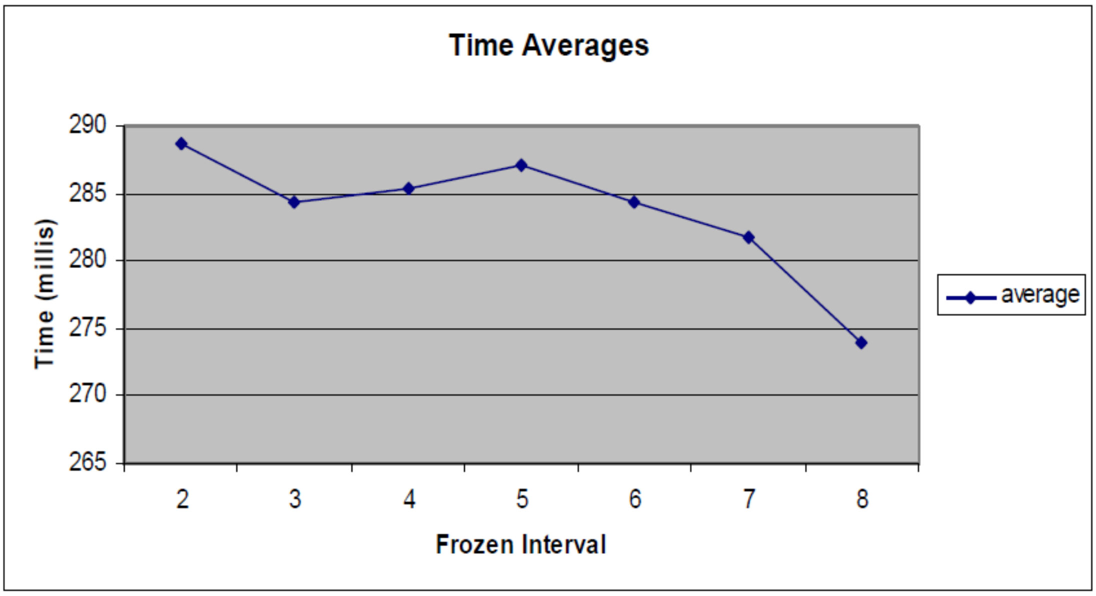


Figure4.13: The Time Averages for “Adaptive Reschedule” method

Figure 4.14 shows better chromosome sizes obtained with adaptive reschedule from thatof the original algorithm that proposed by Chen.

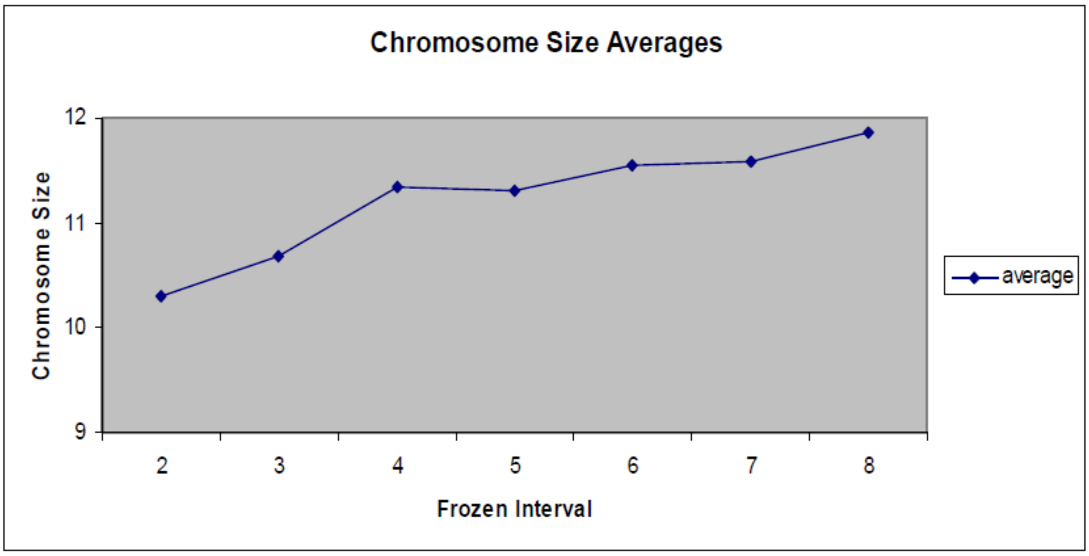


Figure4.14: The Averages of Chromosomes Sizes in the “Adaptive Reschedule”method

Table4.2 shows the enhancement percentage obtained by the Adaptive

Reschedule relative to the results obtained from Chen’s algorithm. The enhancements are clear in terms of the averages of fitness, stability, time and chromosome size.

Metric \ Frozen	2	3	4	5	6	7	8
Fitness	0.25038	0.25481	0.24091	0.25751	0.26571	0.26096	0.2598
Stability	0.73592	0.76106	0.74417	0.75189	0.73110	0.74338	0.74896
Time	0.35976	0.34245	0.40360	0.41531	0.37417	0.30152	0.26283
Chromosome Size	0.30311	0.26428	0.22168	0.2052	0.11697	0.17917	0.14737

Table 4.2: Percentage of enhancement gained by the “Adaptive Reschedule Interval

4.4. Adaptive Frozen Interval:

In Figure4.15, fitness averages are generated for each frozen percent ranges from 0 to 0.9 (of the reschedule interval) when applied to the original algorithm. In other words, if we give the frozen interval value equal to 0.6 multiplied by the reschedule interval then we willget 4160 as a fitness value.

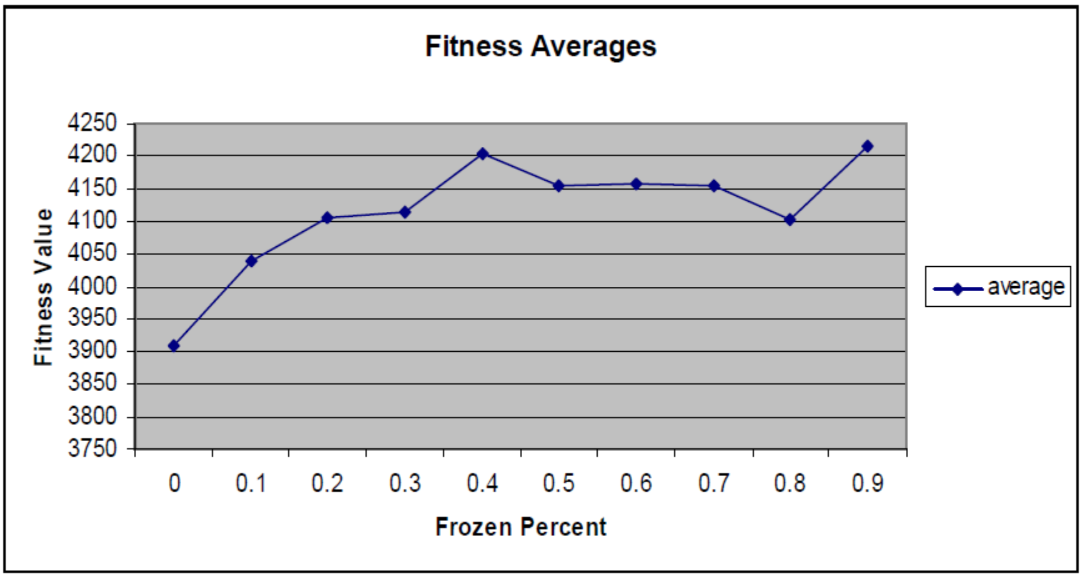


Figure4.15: The Averages of Fitness in the “Adaptive Frozen” method

Figure4.16 shows the stability averages that are obtained for all frozen interval tested

Percent's. We notice that the stability averages are get better while frozen interval converges from the reschedule interval.

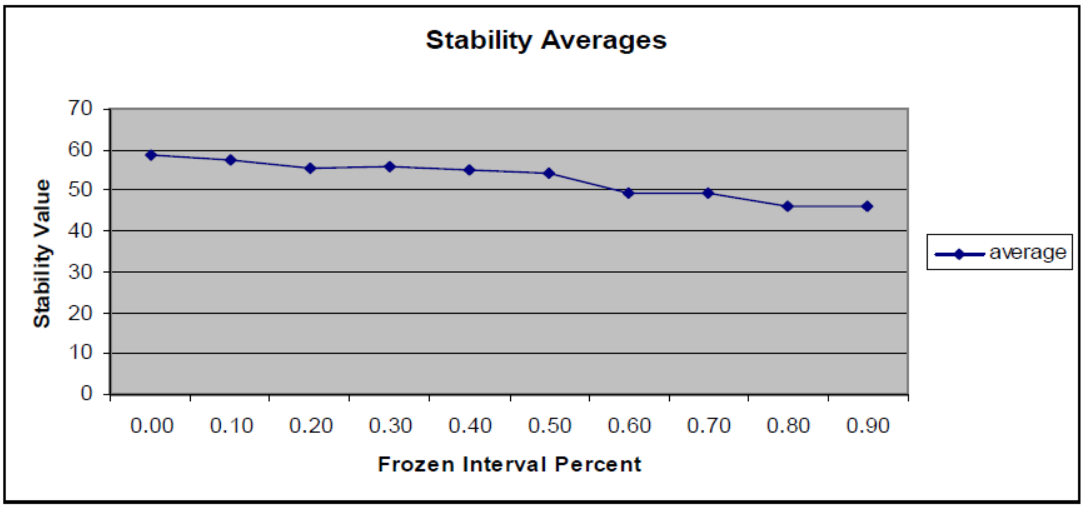


Figure4.16: The Averages of Stability in the “Adaptive Frozen” method

If we are looking for a percent that gives acceptable results as an integrated fitness/stability values, then we can notice that the best percent to choose is 0.4.

4.5. Results Comparisons:

Figure 4.17 shows a comparison between fitness values obtained when applying the different strategies. By studying the figure, we conclude that each algorithm give a real enhancement to the original one. While the best enhancement obtained when making the reschedule interval adaptive with the system current state.

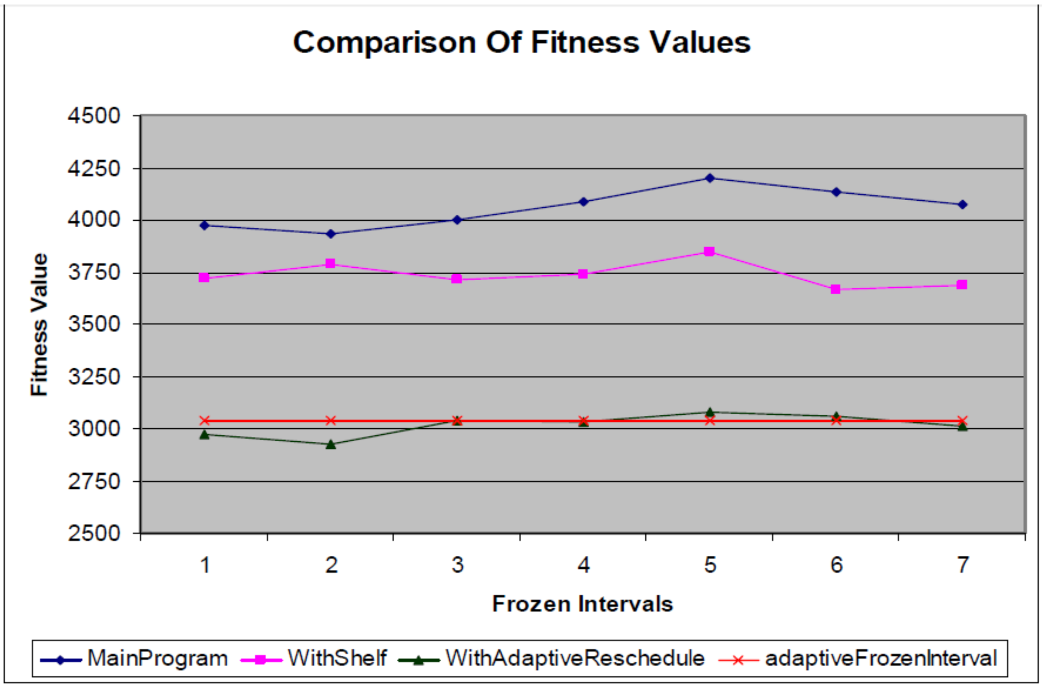


Figure4.17: A comparison of fitness averages obtain by the different algorithms

The Table4.3 below gives a clearer view about the enhancement gained byapplying each of the algorithms. It shows that each of the suggested algorithms results in a clear enhancement on the original algorithm form the fitness value point of view.

These enhancements vary from 0.03 to 0.26 which seems a good enhancement that affects such dynamic environment.

Frozen / Metric	2	3	4	5	6	7	8
On The Shelf	0.06393	0.03608	0.07189	0.08469	0.084282	0.11351	0.094372
Adaptive Reschedule	0.25620	0.24858	0.24084	0.25751	0.265715	0.26096	0.259898
Adaptive Frozen = 0.4 * reScheduleInterval	0.23410	0.22626	0.24017	0.25586	0.275666	0.26475	0.253133

Table 4.3: Summary for the percent of fitness enhancement gained by applying each suggested algorithm to the original one

Figure 4.18 shows a comparison between stability values obtained when applying thedifferent strategies. Each applied algorithm give a big enhancement to the original one.

While the best enhancement obtained when making the reschedule interval adaptive with the system current state.

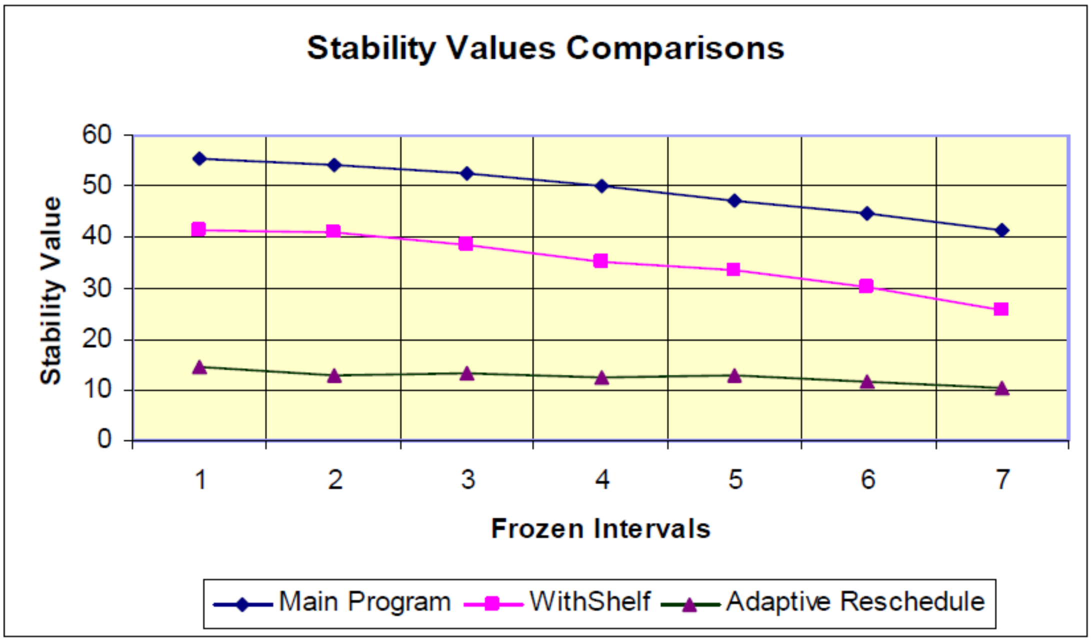


Figure4.18: A comparison of stability averages obtained by the different algorithms

table 4.4 below gives a clearer view about the enhancement gained by applyingeach of the algorithms. It shows that each of the suggested algorithms results in a clear enhancement on the original algorithm form the fitness value point of view. These enhancements vary from 0.24 to 0.75 which is a great enhancement that affects the shop floor stability in a very dynamic production system.

Algorithm / frozen	2	3	4	5	6	7	8
On The Shelf	0.2563	0.2468	0.2663	0.2995	0.2904	0.3230	0.3752
Adaptive Reschedule	0.7359	0.7610	0.7441	0.7518	0.7311	0.7433	0.7489

Table 4.4: Summary for the percent of stability enhancement gained by applying thesuggested algorithm on the original one

Table 4.5 compares all algorithms from the chromosomes sizes point of view. Clear, shorter chromosomes gained by applying our proposed methods.

Algorithms	2	3	4	5	6	7	8
Original	14.78	14.53	14.57	14.23	13.6	14.19	14.91
On The Shelf	12.85	12.79	12.6	12.24	11.46	12.18	11.92
Adaptive Reschedule	10.3	10.69	11.34	11.31	11.55	11.59	11.86
Adaptive Frozen	14.38	14.81	14.71	14.76	13.38	12.9	14

Table 4.5: Average Chromosomes Sizes Comparison

Figure 4.19 shows the comparison of the time metric between all algorithms. Chen algorithm appears by the name of Main Program. Satisfactory results are gained.

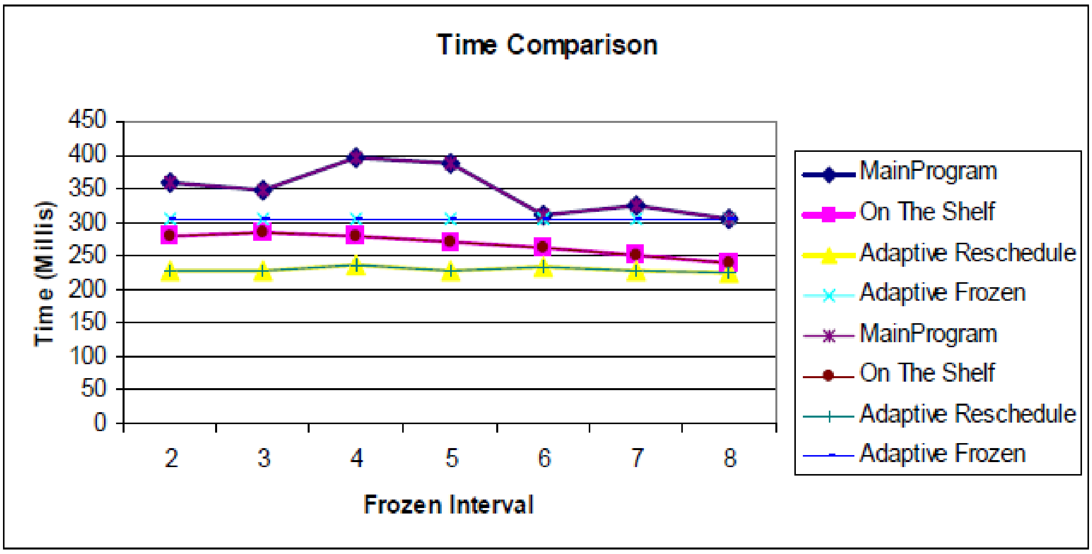


Figure 4.19: Time Comparison

4.6. Multi-Agent Results:

When applying the original algorithm on a MAS system no better results achieved;

Fitness values average equals 3138.7 and time needed is too large, 11108.63 due to the communication time and agents processing speed. Results are shown in Table8.

Frozen	Fitness	Chromosome Size	stability
2	3008.52632	8	27
4	2441.88	8	28
5	2717.88462	8	31
6	2597.86047	7	20

Table 4.6: Output Of Multi-Agent with 2 Agents

Better results expected to be achieved when using heterogeneous agents that defer intheir way of solving the problem by for example using different genetic parameters such as the genetic operations used and the probability of each operator. This difference increases the probability of reaching near optimal fitness value.

Efficiency is enhanced by obtaining better results in terms of fitness, chromosome size, time and stability of the shop floor.

Chapter 5
Conclusion

5.1 Conclusion

Beginning should note that most of the studies related to the field of artificial intelligence are recent studies and the sources available on the World Wide Web is relatively low compared with the areas of computer and other fields, in general, and perhaps this explains the difficulty of finding ideal solutions and the absolute most of the problems related to this field, especially problems related to multi-agents and howget the ideal distribution to different resources, but at the same time we must not underestimate the value of the efforts of the specialists and researchers from the early nineties to the present day,where they have to find solutions and technologies have helped mankind in many things of their lives, among them an example is not limited to sending agent with certain characteristics to measure temperatures in the other planets, where it is illogical to send a man out there, and the examples that we see on a daily basis is the lighting systems inmodern cars where the lighting automatically when it gets dark, because these cars have the sensors in the environment around the sensor and therefore you take a particular action which either will be lit at night or remain as it is in the daytime.

The given results in tables and graphics previous note that the new methodology shows a clear improvement in the results compared with chens algorithm In this sense we urge researchers in this field to intensify their studies to try to find solutions to an absolute 100%

despite the difficulty of it being studies of this type dependent random inputs have, but we are trying to reach the output represents a fixed line or close to it as much as possible

5.2. Future Directions

Until now no good results obtained by applying the algorithm on a MAS, but many ideas that exist make a better use of MAS to affect the results positively. Some of these ideas are:

• Using heterogeneous agents instead of homogeneous ones.

• Distribute the problem between the agentsso that every agent has a small chromosome compared to the one associate with the problem as a one block.

• To test the presented algorithms on Make toStock (MTS) factory instead our proposed make To Order (MTO).

• Finally, try to make the frozen interval adaptive with the mean of communication time needed by agents to finish their assigned work, and thus no wasted time results from waiting this agent to submit their results.

References:

[1]A. Auyeung, I. Gondra, and H. Dai (2003).Multi-Heuristic List Scheduling Genetic Algorithm for Task Scheduling, Proc. 2003 ACM Symp. Applied Computing, pp. 721-724.

[2]Almeida, M. R., S. Hamacher, M. A. C. Pacheco, and M. B. R. Velasco (2001).Applying Genetic Algorithms to the Production Scheduling of a Petroleum Refinery. In MIC'2001 - 4th Metaheuristics International Conference, pp.773-777.

[3]Al Globus, James Crawford, Jason Lohn, and Robert Morris (2002).Scheduling Earth Observing Fleets Using Evolutionary Algorithms: Problem Description and Approach, Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space, Houston, Texas.

[4]Anderson, A production system theory of serial memory, OCT2010, Psychological Review, Vol 104(4), 728-748.doi: 10.1037/0033-295X.104.4.728

[5]Bartk, R.(October 2011). Conceptual Models for Combined Planning and Scheduling. Preceding the eCP9 Post-Confrerence workshop on Large Scale Combinational Optimization and Constraints, Alexandria, USA, pp.2 – 14

[6]Bradley J. Clement, Anthony C. Barrett, (2002). Coordination Challenges for AutonomousSpacecraft, AAMAS-02 workshop notes on Toward an Application Science. MAS Problem Spaces and Their Implications to Achieving Globally Coherent Behavior.

[7]BartákRoman (2002). VisoptShopFloor.On the Edge of Planning and Scheduling. CP 2002, pp.587-602

[8]Barták Roman (June 2000). A Slot Representation of the Resource-Centric Models for Scheduling Problems, in Proceedings of ERCIM Working Group on Constraints/CompulogNet Area on "Constraint Programming" Workshop, Padova.

[9]Bierwirth, C., Kopfer, H., Mattfeld, D. C., and Rixen, I. (1995). Genetic algorithm based scheduling in a dynamic manufacturing environment. In Palaniswam, M., editor, Proceedings of the Second Conference on Evolutionary Computation, pp.439-443, IEEE Press, New York, New York.

[10]BRAUNE R., WAGNER S., AFFENZELLER M. (2004). Applying Genetic Algorithms to the Optimization of Production Planning in a Real-World Manufacturing Environment, Cybernetics and Systems, Austrain Society for Cyberretic Studies, pp.689-694, Seite(n)41–46.

[11]E.K. Burke, D.G. Elliman, R.F. Weare (2012).A Genetic Algorithm Based University Timetabling System, East-West Conference on Computer Technologies in Education.rimea, Ukraine, 2012, pp. 35-40.

[12]Charypar, D. (2005). Generating complete all-day activity plans with genetic algorithms, Transportation – Amsterdam – 2005 Vol. 32 Num. 4, pp.369-397.

[13]Chen, K. J. P. Ji (2007). A genetic algorithm for dynamic advanced planning and scheduling (DAPS) with a frozen interval. Expert Syst. Appl. 33(4), pp.1004-1010, [DBLP:journals/eswa/ChenJ07].

[14]Cheng et al (Dec. 2011).Production Scheduling of Continuous Flow Lines. Multiple Products With Setup Times and costs, Production and Operation Management, vol. 7, No.4.

[15]Das S., P. Gonsalves, R. Krikorian and W. Truszkowski, (1999).Multi-agent planning and scheduling environment for enhanced spacecraft autonomy. Artificial Intelligence, Robotics and Automation in Space, Preceding of the fifth international symposium, ISAIRAS 99, held 1-3 June 1999 in ESTEC, Noordwijk, the Netherlands. Edited by M.Perry.ESA SP-440. Paris: European Space Agency, pp.91.

[16]DiptiSrinivasan and G. B. Tettamanzi (2011). An integrated framework for devising optimum generation schedules. In ICEC'95 [1251], pp.1-4.yprog. ga95aSrinivasan.

[17]Do, M. and Kambhampati, S. (2001). Sapa: A domain-independent heuristic metric temporal planner. In Proceedings of the Sixth European Conference on Planning (ECP-01), pp.109–120.

[18]Edmud F. Watson, Deborah J. Medeiros, Randall P. Sadowski(2012). A Simulation-Based Backward Planning Approach for Order-Release, Winter Simulation Conference 1997, pp.765-772

[19]Edwin A. Kjeldgaard , Dean A. Jones, George F. List, Mark A. Trnquist (2011). Planning and Scheduling for Agile Manufacturers: The Pantex Process Model, SANDIA REPORT, SAND98-0030 . UC-2070, Unlimited Release, Printed February 1998.

[20]Gerard GaalmanDepartment of Production Systems Design, Faculty of Management and Organization, (2010)University of Groningen, P.O. Box 800, 9700 AV Groningen, Netherlands.

[21]Goldberg, D.E., (1989). Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley.

[22]Haddawy, P. and Hanks, S. (1998). Utility models for goal-directed, decision-theoretic planners.Computational Intelligence, 14(3) pp.392–429.

[23]H-L Fang, P.M.Ross and D.Corne (2010). A Promising Hybrid GA/Heuristic Approach for Open-Shop Scheduling Problems, in Proceedings of ECAI 94: 11th European Conference on Artificial Intelligence, A. Cohn (ed), pp.590-594, John Wiley and Sons Ltd.

[24]Holland, J. H. (1975). Adaptation Natural and Artificial Systems.University of Michigan press, Ann Arbor

[25]J. F. Gonçalves, J. J. M. Mendes, M. G. C. Resende (November 2005). A hybrid genetic algorithm for the job shop scheduling problem, European Journal of Operational Research, Volume 167, Number 1, pp. 77--95 - November 2005.

[26]Kevin R. GUE, George L. NEMHAUSER, and Mario PADRON (2010). Production Scheduling in Almost Continuous Time, IIE Transactions 29:5, pp. 391-398.

[27]Han.L, G. Kendall, and P. Cowling (November 2002). An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem.In Proceedings of the4th AsiaPacific Conference on Simulated Evolution and Learning (SEAL'02), pp.267-271.

[28]James Cunha Werner and Terence C.Fogarty (2001). Map algorithm in routing problems using genetic algorithm. Technical report, London South Bank University, SCISM, South Bank University, 103 Borough Road, London SE1 0AA.

[29]J.A. Ottjes, H.P.M. Veeke (April 2000).Production Scheduling of Complex Jobs with Simulation. In M. Ades (eds.); Proceedings of the Business and Industry Simulation Symposium (ASTC 2000), advanced simulation technologies conference (Washington D. C.), society for computer simulation international, SanDiego, California, p.157-162. ISBN 1-56555-199-0.

[30]K. Kurbel, B. Schneider and K. Singh (2009). Parallelization of Hybrid Simulated Annealing and Genetic Algorithm for Short-term Production Scheduling, in Proceedingsof International Conference on Intelligent, Knowledge and Integration forManufacturing, Nanjing, China, March 28-31, pp321-326.

[31]K. Poeck and C. Hestermann (2012). An intelligent interactive support system for shortand mid-term planning in production planning and control system.In N.

[32]Lee, Y. H., Jeong, C. S., & Moon, C. (2002).Advanced planning and scheduling with outsourcing in manufacturing supply chain.Computers and Industrial Engineering,43(1–2), pp.351–374.

[33]M. Sadeh (ed.):Intelligent Manufacturing Systems, pp.285--288. Working Notes of the Workshopheld at the 14th IJCAI-95, August 19.

[34]Mathijs de Weerdt, Adriaanter Mors and CeesWitteveen. (2005). “Multi-agent Planning: An introduction to planning and coordination”, In Handouts of the European AgentSummer School.

[35]Mathijs de Weerdt and CeesWitteveen (2006).Multiagent planning: problem properties that matter. In Ed Durfee and David Musliner (Eds.).Proceedings of the AAAI SpringSymposium on Distributed Plan and Schedule Management, pp. 155--156.

[36]Mathijs M. de Weerdt (2003). Plan Merging in Multi-Agent Systems.PhD Thesis, Delft University of Technology, Delft, The Netherlands.

Michell, M. , (1997). An introduction to genetic algorithm s.The MIT Press, Cambridge, Massachusetts.

[37]M. K. Lim and Z. Zhang(2002). Iterative multi-agent bidding and co-ordination based on genetic algorithm. Proceeding of 3 Complex Systems, and E-Businesses, Erfurt, 7-10October 2002, pp. 682--689.

[38]Patig, S. (2001).Using planning steps for production planning in.Proceedings of the International NAISO Congress on Information Science Innovations, ISI 2001, Dubai(U.A.E.). pp.516

[39]Penberthy, J. and Weld, D. S. (1994).Temporal planning with continuous change.In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), pp.1010–1015.AAAI Press, Menlo Park, CA.

[40]Smed, Johnsson, Johtela, and Nevalainen (2003). Techniques and Applications of Production Planning in Electronics

Manufacturing Systems, in C. T. Leondes (ed.), Manufacturing Processes, volume 5 of Computer Aided and Integrated Manufacturing Systems, pp. 1–48, World Scientific Publishing, New Jersey, NJ.

[41]Smith, D. E. and Weld, D. S. (1999).Temporal planning with mutual exclusion reasoning.In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99).Morgan Kaufmann Publishers, San Mateo, CA, pp.326-333.

[42]Stöppler, S. and Bierwirth, C. (1995).The Application of a Parallel Genetic Algorithm to the n/m/P/CmaxFlowshop Problem, < http: // medusa.logistik.uni-bremen.de / Papers /Arbeitsberichte.html >.

[43]T. J. Harding and N. J. Radcliffe, Quadstone Ltd. and P. R. King, BP Exploration Harding, T.J., Radcliffe, N.J., and King, P.R (1996).Optimisation of Production Strategies UsingStochastic Search Methods, paper SPE 33518, Proc., 1996 NPF/SPE European 3DReservoir Modeling Conference, Stavanger, 16–17 April.

[44]Tuson, A L (2011). The implementation of a Genetic Algorithm for the scheduling and topology optimisation of chemical flowshops, Technical report TRGA94-01, Physical Chemistry Laboratory, Oxford University, U.K.

[45]Wan A.D.M. and Braspenning P.J. (1996). Limits to Ground Control in Autonomous Spacecraft.Telematics and Informatics 12 (3/4), pp. 247-259. Also appeared in NASA Conference Proceedings: NASA CP-3296, 1995.

[46]Wiers, V.C.S., and T.W. van der Schaaf, T.W. (2012) A framework for decision support in production scheduling tasks, Production Planning & Control, 25:2, pp.533-544.

[47]Wiers W.C.S. (1997). A review of the applicability of OR and AI scheduling techniques in practice. Omega International Journal of Management Science, Vol.25, N2, pp.145-153.

[48]Wolfman, S. A. and Weld, D. S. (2001).Combining linear programming and satisfiability solving for resource planning. Knowledge Engineering Review, 16(1): pp.85–99.