AN EFFICIENT AND FLEXIBLE INTEGRATED PLANNING AND SCHEDULING SYSTEM FOR MULTI-AGENT SYSTEMS USING EVOLUTIONARY ALGORITHMS

By
Hutaf Salah Al-Deen Natoureah

Supervisor
Dr. Nadim Obeid, Prof

Co-Supervisor
Dr. Imad Salah

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

Faculty of Graduate Studies
The University Of Jordan

August,2007

SIGNATURE PAGE

DEDICATION

I dedicate this work to every one who supported me while I was writing this thesis; to my parents for there continues encouragement and understanding. A special and warm thanks to my teacher Maha Abu Rumailah who's without her great help and support I never could complete my high studies. Of course to my close and sweet friends: Huda, Rana, Shaden, Bayan, Maha, Hama, Dema and Yousaira. Finally, beautiful thanks to the European Information Technology Center (EITC) Center staff for their kind feelings and supports.

AKNOWLEDGEMENT

First, thanks to GOD, who helped me with strength, faith and capability to accomplish my thesis. Secondly, I owe a lot to my supervisors Prof. Nadim Obaid and Dr. Imad Salah for their continuous support, supervision and understanding.

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

LIST OF ABBREVIATIONS

Number	Abbreviation	Refers to
1	AI	Artificial Intelligence
2	MAS	Multi-Agent System
3	DAPS	Dynamic Advanced Planning Scheduling
4	DAI	Distributed Artificial Intelligence
5	DPS	Distributed Problem Solving
6	OSSP	The Open-Shop Scheduling Problem
7	GA	Genetic Algorithms
8	RCCP	Rough-Cut Capacity Planning
9	MRP	Material Requirements Planning
10	CRP	Capacity Requirements Planning
11	IEFPSS	Integrated Efficient and Flexible Planning and Scheduling System
12	FMS	Flexible Manufacturing System
13	DAI	Distributed Artificial Intelligence
14	MTO	make-to-order
15	MRP	Material Requirement Planning
16	PCB	Printed Circuit Boards
17	ELSP	Economic Lot Scheduling Problem
18	PGA	Parallel Genetic Algorithm
19	PRSA	Parallel Recombinative Simulated Annealing)
20	WIP	Work-In-Process
22	MTS	Make To Stock

LIST OF ALGORITHMS

AN EFFICIENT AND FLEXIBLE INTEGRATED PLANNING AND SCHEDULING SYSTEM FOR MULTI-AGENT SYSTEMS USING EVOLUTIONARY ALGORITHMS

By
Hutaf Salah Al-Deen Natoureah

Supervisor
Dr. Nadim Obeid, Prof

Co-Supervisor
Dr.Imad Salah

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. Additionally, there has been a growing interest in agent-oriented problem solving, which provides the basis of our proposed methods that adds efficiency and flexibility to the process of planning and scheduling of the various production systems.

This thesis proposes methods are applied on a planning and scheduling system based on Evolutionary algorithms to provide a solution that concerns finding an efficient and flexible integrated planning and scheduling system for a Multi-Agent System(MAS). These methods aims to solve the problem of finding the most efficient plan and schedule combination for a continual and unstable production system while maintaining system flexibility.

We utilized previous method to achieve more efficient and flexible system that integrates planning and scheduling solution in a production system by introducing different enhancement methods and then solving the problem using MAS. A MAS consists of a set of entities with specific properties and capabilities (especially those concerning solving the problem in hand). These agents may communicate and cooperate with each other to achieve better solution as possible while maintaining some pre-specified constraints. The three proposed methods are: “On the Shelf” and “Adaptive Reschedule Interval” and “Adaptive Frozen interval”, in order to increase the efficiency of the planning and scheduling in manufacturing. These methods are applied first on a single then to a multiple agents system. A five-level product structure is adopted to test our strategies.

Results show better fitness and thus less cost, more stability, less average chromosome size and less time get after applying the proposed methods. All These metrics will be discussed later in the thesis.

INTRODUCTION

1.Problem Description

Agent-oriented problem solving strategy gained a high interest last years. This strategy deploy 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). An intelligent agent distinguished by having knowledge about the problem space and how to 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 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 which 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 account, anticipated future developments. 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.

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 that the 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.

"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).

"Planning / scheduling involve determining when to perform which activities as the production system capabilities (e.g. machines capacity)", (Bradley J. Clement 2002).

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

- A set of autonomous agents with their assumed capabilities.
- A set of the available machines with their assumed setup times, capacity and the sub products that each one produce.
- A production structure that shows all products and sub products produced by the manufacture and their precedence relations.
- 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.
- A set of initial conditions, which describes the system current state.

Planning and Scheduling problem is considered as NP-Hard problem, 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 known evolutionary algorithms to plan and schedule the production system, thus achieving near optimal results.

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

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).

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.

2. Refinement and Extended Planning

"Process planning is the systematic determination of the detailed methods by which parts can be manufactured from raw material to finished products",(Smed 2003).

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).

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 equipments, these machines can be programmed to do some processing operations under the direction, (Smed 2003).

To find a sequence of actions that move the system from its current state to a pre-specified 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).

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 utility maximization) (Haddawy 1998), limited resources (Wolfman 2001), and planning under uncertainty.

From this view of complexity in planning such systems, we suggest using Evolutionary Algorithms as an optimization techniques that 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

3.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 define 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).

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).

4. 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, aeroplane landing scheduling, train scheduling and production scheduling.

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.

5. 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 concedration 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).

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.

6. 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:

- A multi-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.
- Evolutionary algorithms are defined and proposed as to be the adequate and most appropriate solution to the problem above.
- Genetic Algorithms are provided as the evolutionary algorithm that chose to be used in solving this problem.
- 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.
- Several methods are proposed to add efficiency and flexibility to the studied manufacturing system.
- 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.
- Experimental results show the flexibility and efficiency of our proposed framework.

- 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 in (Lee 2004).

7. 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. Then, a discussion of the problem theory and the solution adopted by Chen, followed by discussing our proposed enhancement methods. In the last of Chapter3, a complete example of the problem is discussed. An analysis of the results is discussed in Chapter4. Finally, we list our conclusion with a spot of light over some future improvements.

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.

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. 1999) gave a survey of possible conceptual models for scheduling problems with some planning features. A comparison is done to find their advantages and disadvantages. Furthermore, analysis of the problems behind industrial planning and scheduling is done after a study within the VisOpt project whose task is to develop a generic scheduling engine for complex production environments.

Mathijs de Weerdt (Mathijs 2003) was interested in finding a way to coordinate planning agents without revealing all vital information bi 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. 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.

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. 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) planning approach 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.

(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.

Poeck (K. Poeck 1995) proposed a support system that covers the whole range from completely interactive scheduling and rescheduling to totally automatic plan generation.

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. The aim is to cope with uncertainty in production planning.

2. Multi-agent Scheduling

Cheng (Cheng et al Dec. 1998) 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 anther. 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. 1997) addressed the problem of allocation of tasks between scheduling systems and human schedulers for various types of production units.

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 represent a single production task to be processed on a specific machine (processing machine or an assembly machine). 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 1997) 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

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 Pantex plant at the US Department of Energy facility. The two production processes are characteristic of flow-shop and job-shop operations.

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.

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. The process of job assignment to machines and the process of handling the negotiation between the different autonomous agents is 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.

Dipti Srinivasan (Dipti Srinivasan 1995) 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 non linear 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.

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 genetic algorithm. 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.

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.

Andy Auyeungn (A. Auyeung 2003) attempted to solve the problem of multi-processor 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 1994) presented a Genetic Algorithm Based University Timetabling scheduling System.

Lang and Ross (H-L Fang 1994) improved the previously best known results produced by tabu search on some benchmarks Open-Shop Scheduling Problems (OSSPs). A hybrid Genetic Algorithm is used that makes the system more flexible and easy to use in terms of development time.

Andrew (Tuson, A L 1994) presented an implementation of a genetic algorithm to solve the problem of scheduling a production system where there is a n umber 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.

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 o 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 modela static, dynamic and non-deterministic production environments using Genetic Algorithm. This algorithm is tested in a dynamic environment under different workload situations. 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. Kurbel 1995) 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.

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.

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.

PROBLEM DESCRIPTION AND ANALYSIS

Problem Analysis

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.

1.2. Agent Population

Agent population can be characterized 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.

1.3. 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.

Resources

Agent may need to use different resources to accomplish its tasks. Scheduling shared resources (ex. mchines) 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.

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 depends 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.

2.Chen's Algorithm

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 in-hand 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 fro them. This strategy minimizes the number of the rescheduled items and thus adds stability to the shop floor.

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 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 arrive 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.

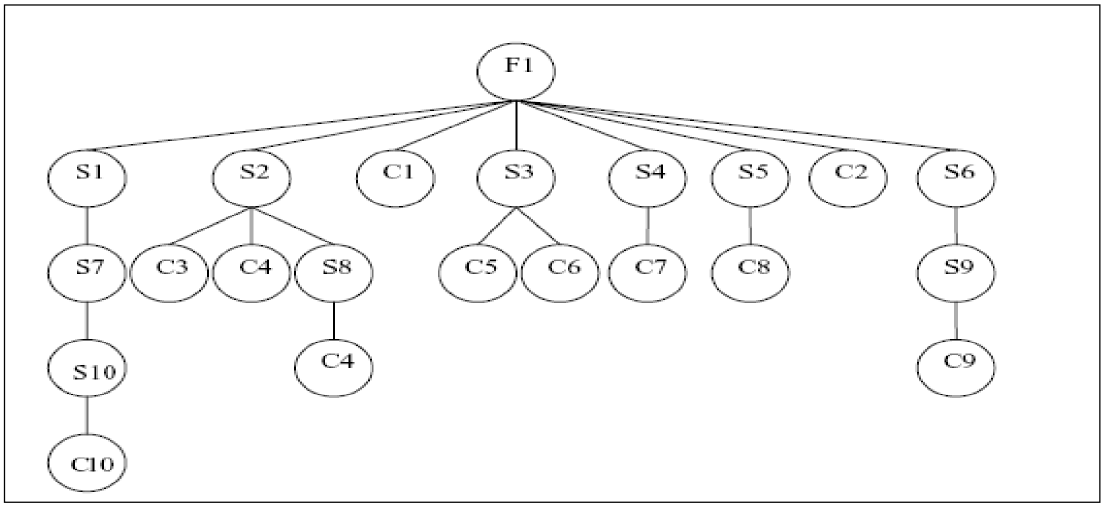


Figure 1: The Five-Level Product Structure

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).

A major advantage of a GA approach is that it is a stochastic-directed-searching 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.

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 rder

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

2.5Population Diversity

It is one f 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).

2.6 Population Size

Population size reflects the number of individuals in a population. The larger the population size, the better the chance that an 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.

2.7 Chromosome 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 : Examaple of chromosome with binary encoding

Chromosome 1: 110010010101010011100110

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 algorithms 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 his 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 Routlette wheel selection operation, and then apply the parameterized crossover operator with probability equals 0.75 to produce two new childs. 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.

```
for each order
  construct the sequence of needed products
end for each order

define a chromosome of length equals to the number of products in the
sequence

// start encoding
construct the chromosome from double genes

for each gene
        associate 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
```

Algorithm 1: Chromosome Encoding and Decoding

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 to their 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 has only 5% probability to be selected. These probabilities appear clearly in Figure2. If we imagine this figure as a wheel and we want to roll this wheel, then for sure, chromosomes with higher fitness (ClassA chromosomes) will have a better chance to be selected.

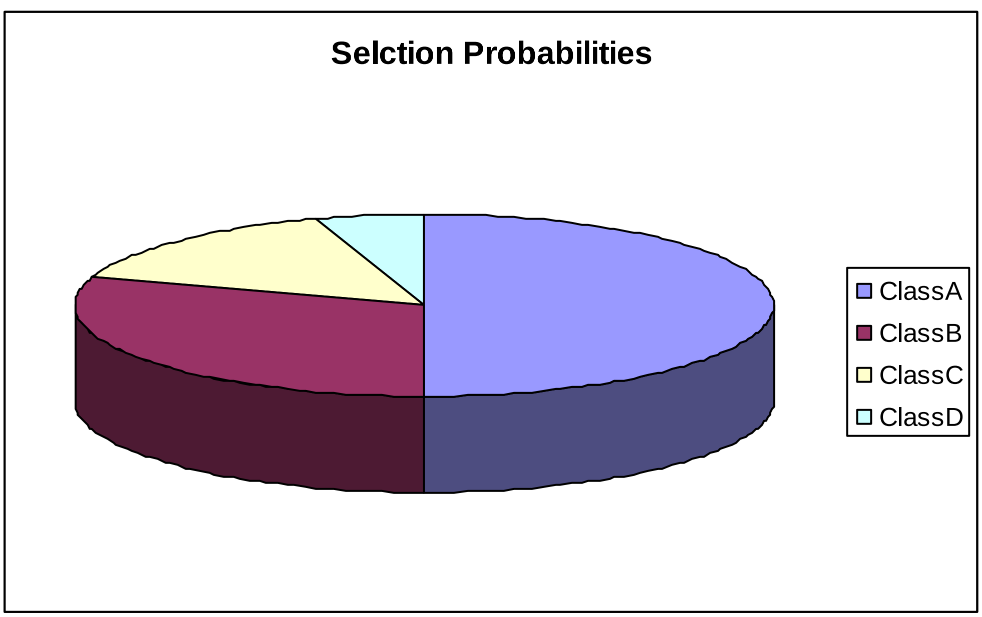


Figure2: Routlette Wheel Selection

There are other types of selection operators such as Sigma Scaling selection and Rank Selection operators. But the Routlette Wheel selection is the most popular one.

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 from the 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:

  Eaxmaple 3 : Example on single-point crossover function

  Parent 1 : 1000110101100111110011000

  Parent 2 : 0011001100111100001010101

  Crossover point : 11

  After Crossover, the new offspring are:

  0011001100100111110011000

  1000110101111100001010101

  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 "position Bias" 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 operation 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.

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.

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.

2.12 Stopping Criteria

The simulation is done for 120 hour (15 working day, for 8-hour working day) and is repeated 100 times 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 are 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.

2.13 The Genetic Algorithm Procedure Summary

The main steps in genetic algorithms are shown in Figure3 below:

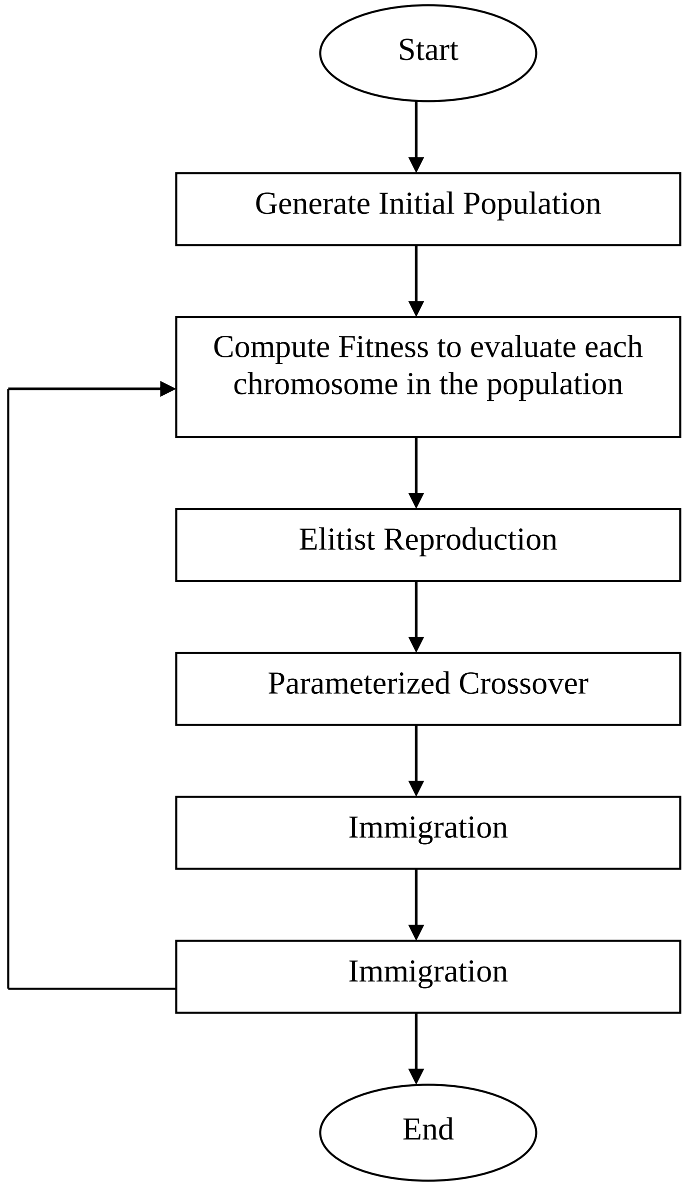


Figure3: The Genetic Algorithm Steps

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 sub-assemblies complete processing. This waiting most of times 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 and where 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.The New 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.

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.

3.2. 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 adaptive reschedule 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 and Equation3 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

$$\mathbf{NRI} = \text{MAX}\ (\text{Max}(P_{i=1..N}) * \text{MAX}(\text{Qnty}_{j=1..NO}), \text{FrozeInt} * \text{CONST}, \text{ORI} - \sum_{c=1}^{M} CT - c.CT \sum - \sum_{r=1}^{M} CT - r.TIH.SPT + [\sum_{y=1}^{RT} y.EPT - CT] / RT) \quad \ldots\ldots\ldots \quad (3)$$

Where:

NRI : New Reschedule Interval

P_i : Processing Time needed to produce product i

N : number of products that can be manufactured by this factory

Qnty_j : Order Qunatity (amount)

NO : Number Of Orders Currently Available

FrozeInt : current Frozen Interval

ORI : Original Reschedule Interval

M : number of Machines

CT : Current Time Slot

c.CT : Current Time inside machine m (the time slot when the machine finish processing its last in hand product)

TIH : Task In Hand (for the machine)

SPT : Start Processing Time (for the task)

RT : Rescheduled Tasks

EPT : End Processing Time (for a task)

CONST : constant (set to 2 after experiments)

Equation 4 states that the new reschedule interval will have the maximum of three values: First, assuming only one order will be requested, then that order will take processing time equals the request product processing time multiplied with its quantity. Now, assuming also that this product is the one that has the largest processing time upon the other products in the product tree, and that the quantity requested equals to the maximum allowed quantity (see Chen's assumptions). Secondly, the reschedule interval must be larger than the frozen interval (x) times, after several experiments we found that it is most suitable to fix x at 2. The third parameter of the max function depends on the idle times spent by the machines waiting the next reschedule point, and the difference in time between the old schedule finish time and the new schedule for the rescheduled products.

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.

4.Performance Measures

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

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.

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 affect 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 the fitness value. This negative effect caused by the delay cost which resulted form 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.

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

Where:

t : current time.

t_i: 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 total deviation 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

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 each new 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.

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.

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 day2 (hour 16 maximum) and O2 requests 2 items of S3 with deadline day1 (hour 8 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 Table1 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 hour does it take to produce one entity of it.

Table 1: machines and processing time required by each product

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

Assume also that machines required to setup every 8 hours. Setup times assumed for these four machines are in Table2 below.

Table 2: Setup times for the machines in the example

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

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 processed product finish time).

CEP: Child End Processing (for each child product of this product in the product tree. That is, it is the pre-requested product for this product 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.

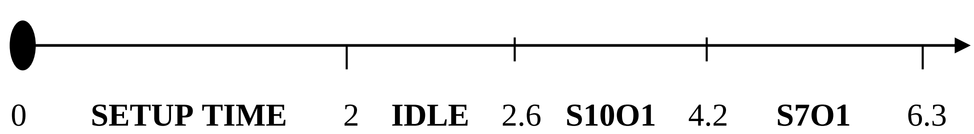
Step6: S7O1 is ready for processing. It will be assigned to machine M2 from the time slot 4.2 until 6.3.Note that M2 spent an idle time between 2 and 2.6 in this schedule. This occur because S7O1 needs C10O1 as a sub-assembly, and this sub-assembly will be finished at 2.6, so that we could not associate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

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:

0.69	0.37	0.41	0.55	0.98	0.22

0.69	0.37	0.41	0.55	0.98	0.22

- 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 fro every gene in the chromosomes.

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.

6. Implementation Language

We implement this strategy and the enhancement methods using JAVA Object Oriented Programming Language (jdk1.5) at windows XP operating system with

1.78 GHz CPU and 256 MB of RAM. A block diagram for these classes is shown in Figure4:

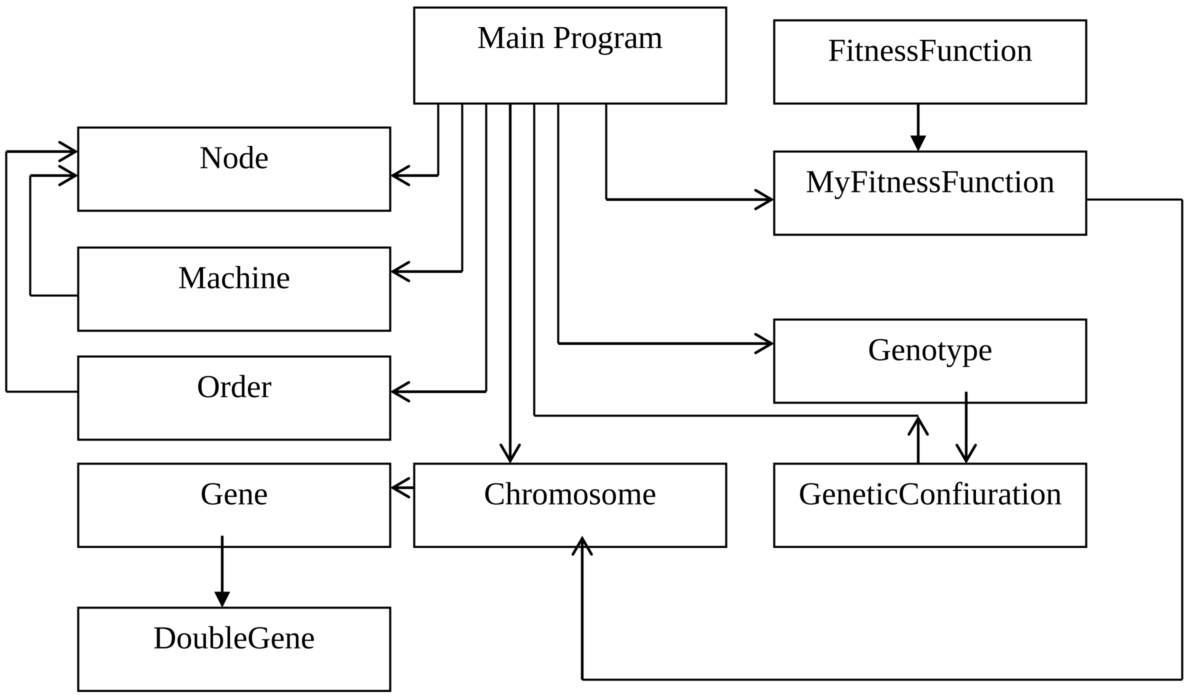


Figure 4: Classes Block Diagram

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 agent production systems.

When applying the Genetic Algorithm, Fitness values will converge gradually to a near optimal value. Below at Figure5 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 Figure6, 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 from generation to another.

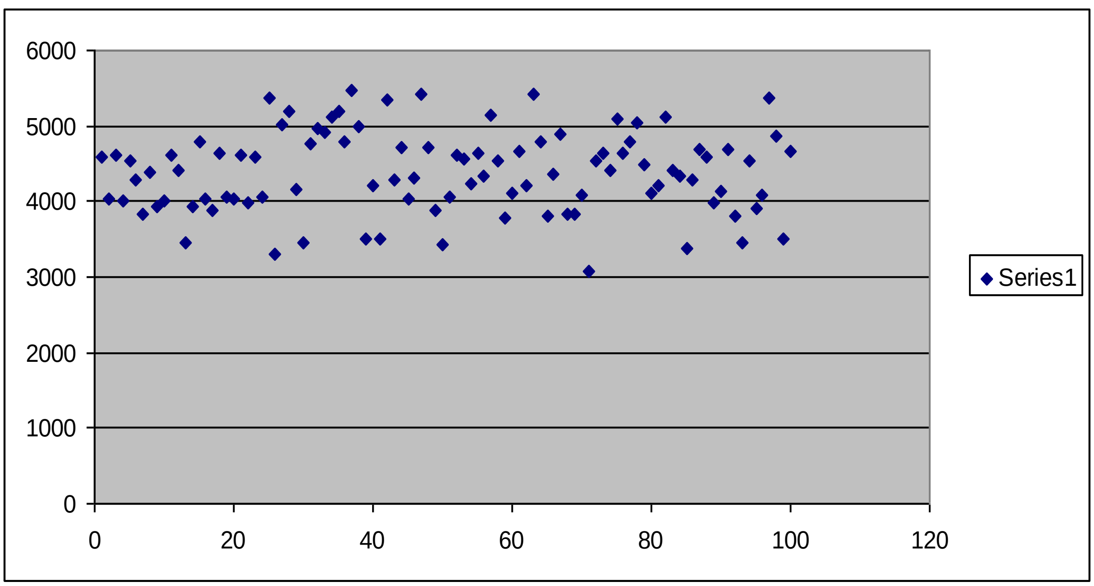


Figure 5: Fitness Values of Chromosomes at the first generation

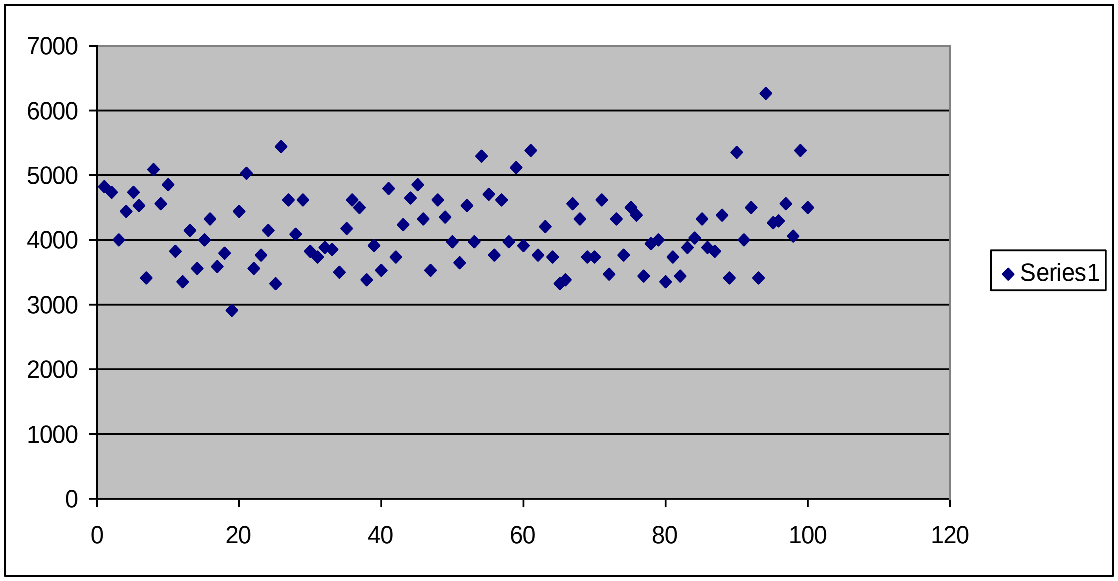


Figure 6: Fitness Values of Chromosomes after 100 generation

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 5 and 30 (step 5).

As mentioned previously, when we talk about the fitness function, the cost of the time that 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 for each 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).

Figure7 shows the fitness averages when applying Chen's algorithms on different frozen 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, this decrement can be explained as an effect to the minimization happened to the number of products that need rescheduling.

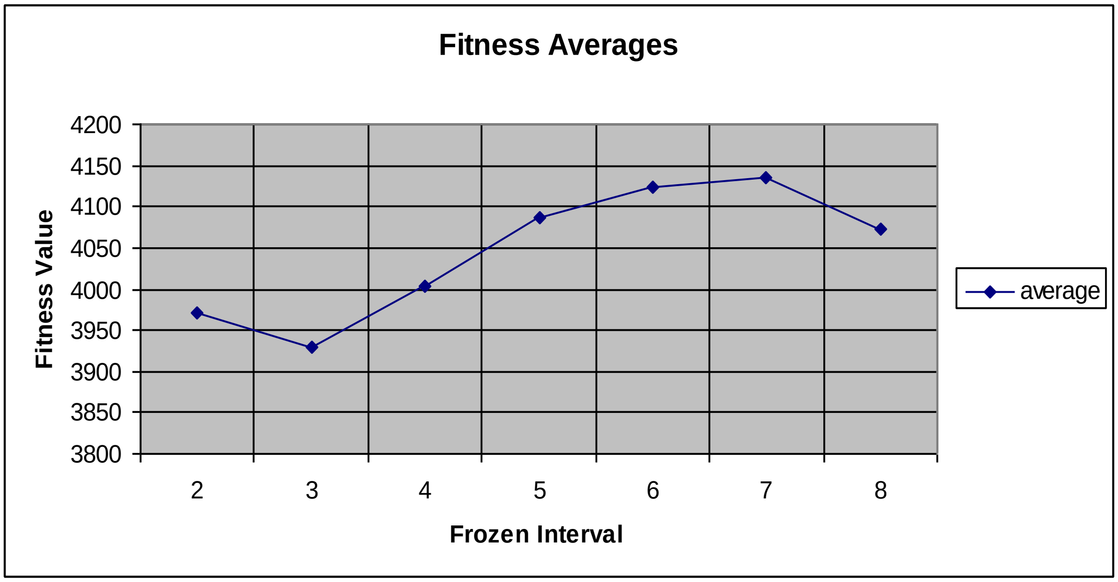


Figure 7: The Chen's algorithm Fitness Chart for 100 Runs

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).

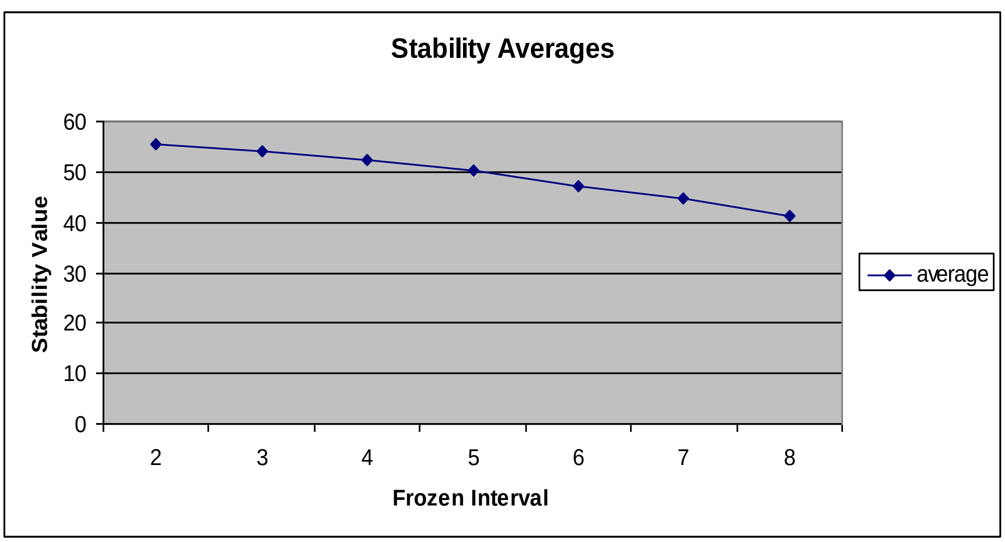


Figure 8: Stability values in Chen's algorithms for 100 run with different frozen intervals

Now, we will study the effect of changing the frozen interval on the average chromosome size. From Figure9, 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 Figure9, we can notice that the average of chromosome size is not decreasing sharply, it is just percents 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 Figure10.

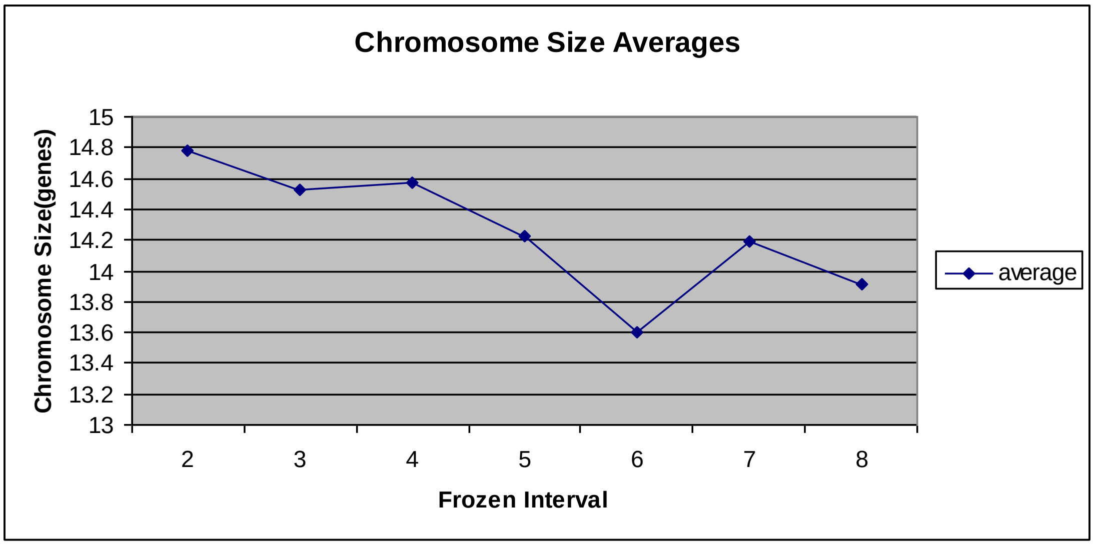


Figure 7: Chromosome Size averages for different frozen intervals in Chen's algorithm

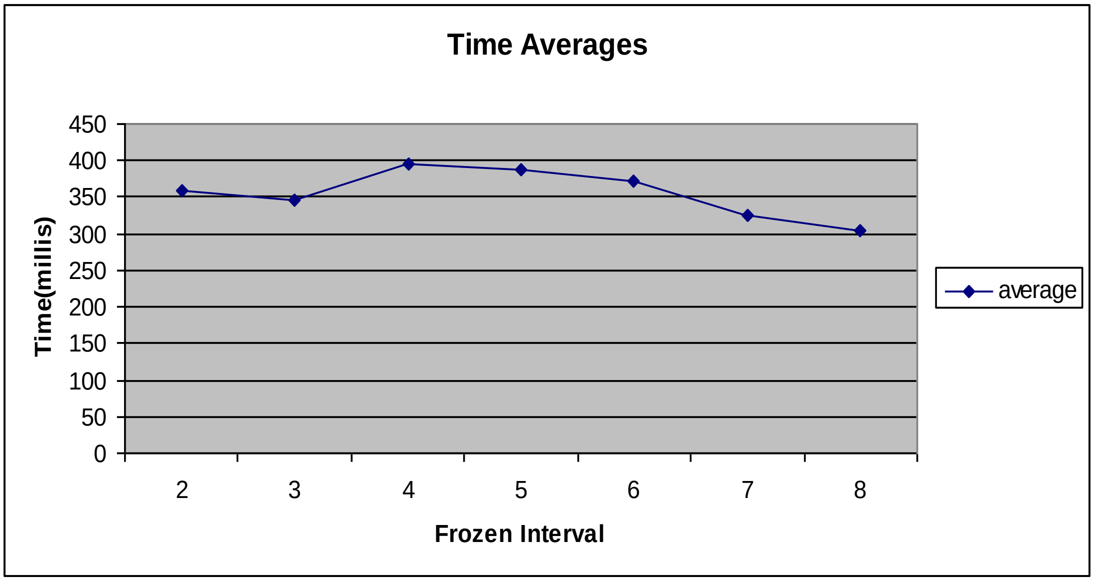


Figure 10: Time averages for different frozen intervals in Chen's algorithm

As a summary, the fitness values are goes to the worse while increasing the frozen interval. 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.

2. On the Shelf Idea Results:

Figure11, 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.

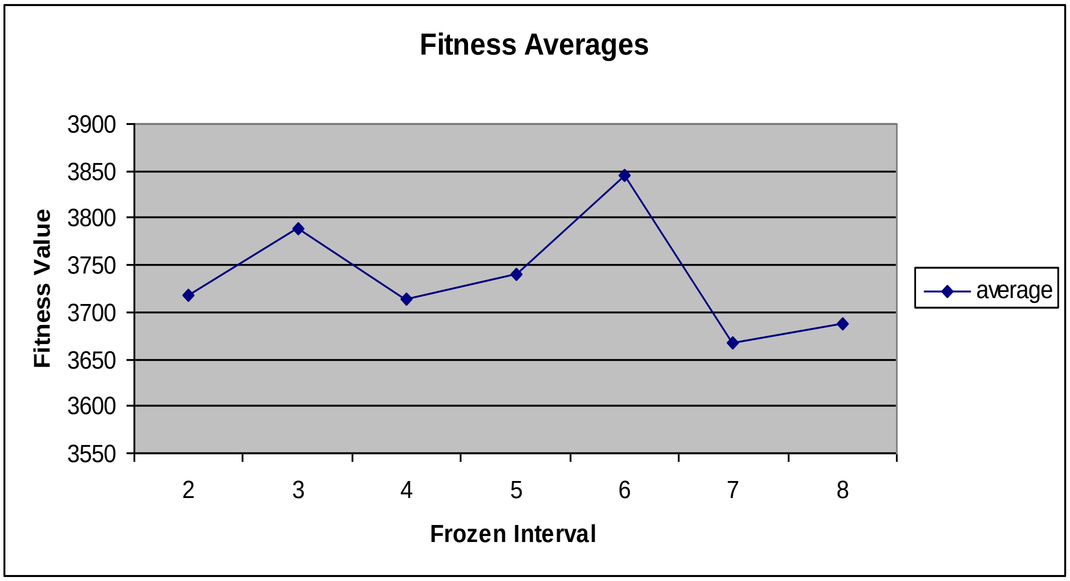


Figure11: The Fitness Values for the“On The Shelf” method

If we look at the stability figure (Figure12), we notice that stability values ranges between 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.

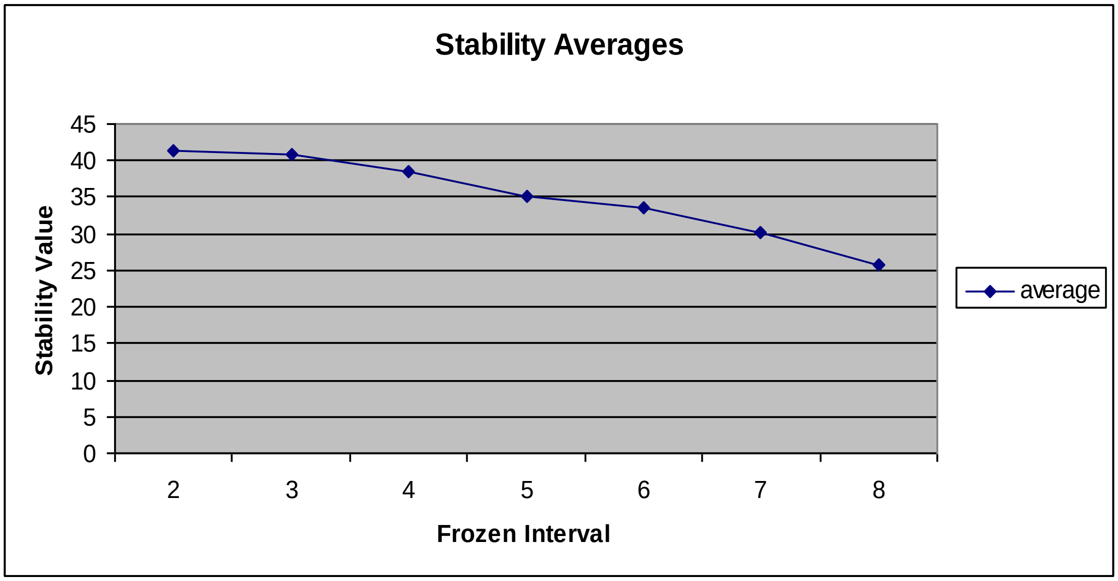


Figure12: Stability Values for the “On The Shelf” idea

When looking at Figure13, 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.

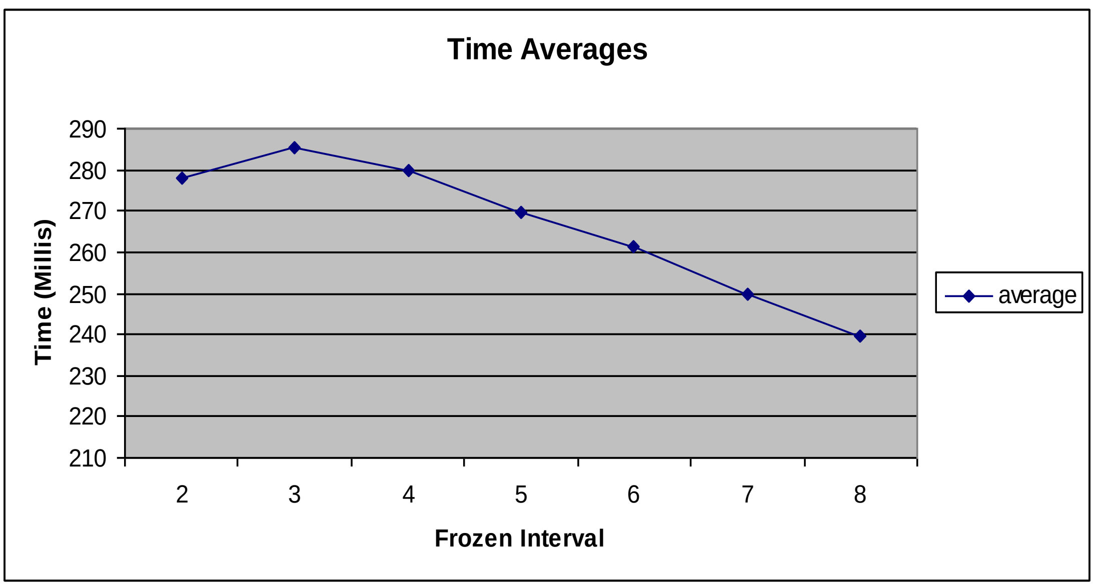


Figure 13: Average of time for different frozen intervals collected from

“On The Shelf” method

Figure14 shows that the average of chromosome size decreased from the average stored 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.

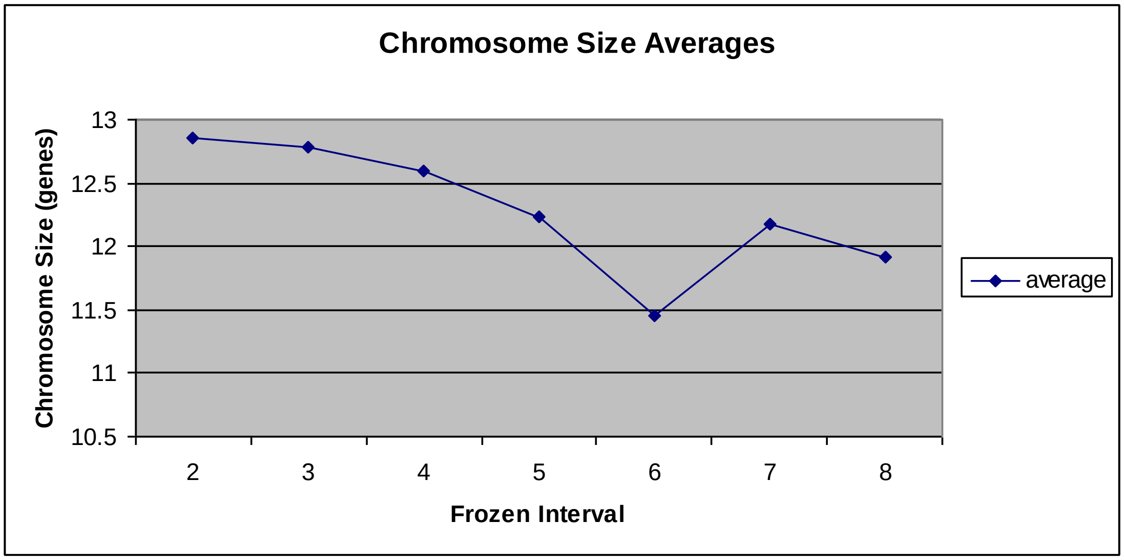


Figure 14: Average of Chromosome Size for different frozen intervals collected from “On The Shelf” method

An enhancement is noticed in all aspects, which is a great enhancement (see Table3). These enhancements percents are calculated by Equation8 below:

Enhancement = (Chen’s result – new result)/ Chen’s result * 100% …………… (8)

Table 3: Enhancement Percentage obtained by "On The Shelf" strategy

Frozen / Metric	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

3. Adaptive Reschedule Interval:

Better fitness values are collected by applying this methodology. As shown in Figure 15 below, Fitness values are better than the results shown in Figure2 (results from Chen).

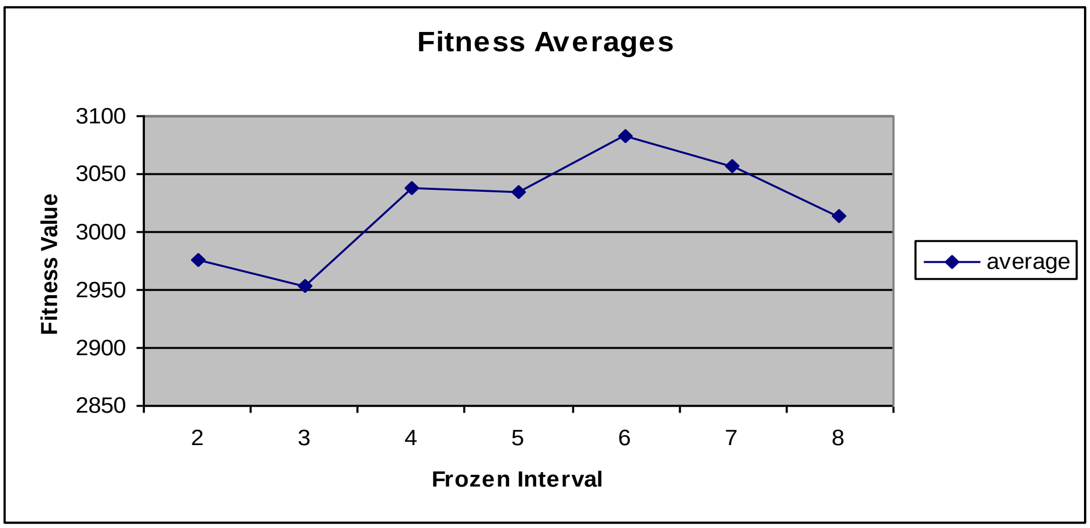


Figure15: The Fitness Values for "Adaptive Reschedule Interval" method

Figure 16, 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.

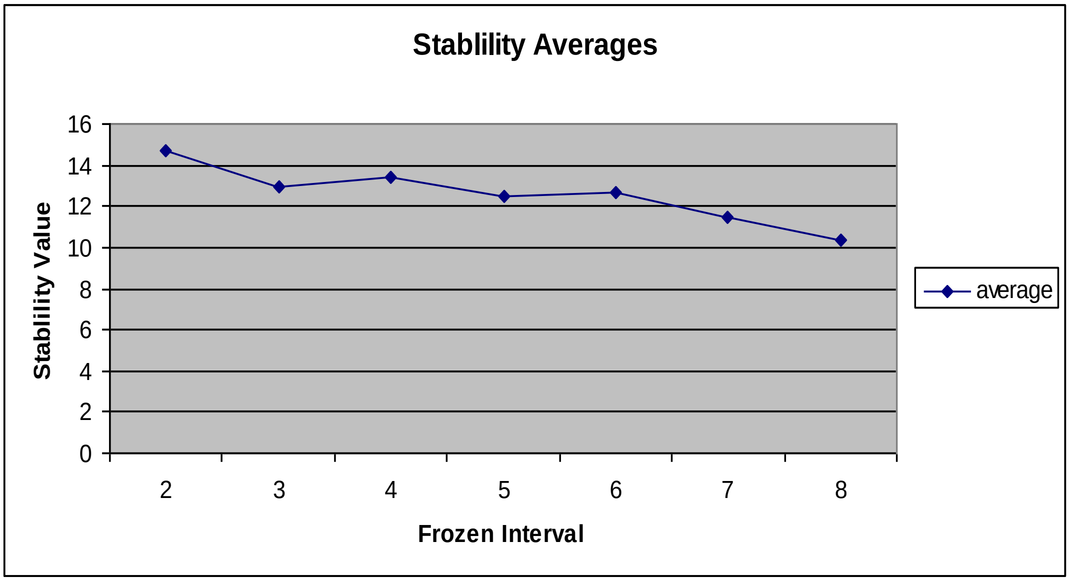


Figure16: The Stability Averages for “Adaptive Reschedule” method

Time averages are show at Figure 17. The same behavior of the time curve in noticed as that in the “On the Shelf” curve, Less time means more dynamicity less time wasting.

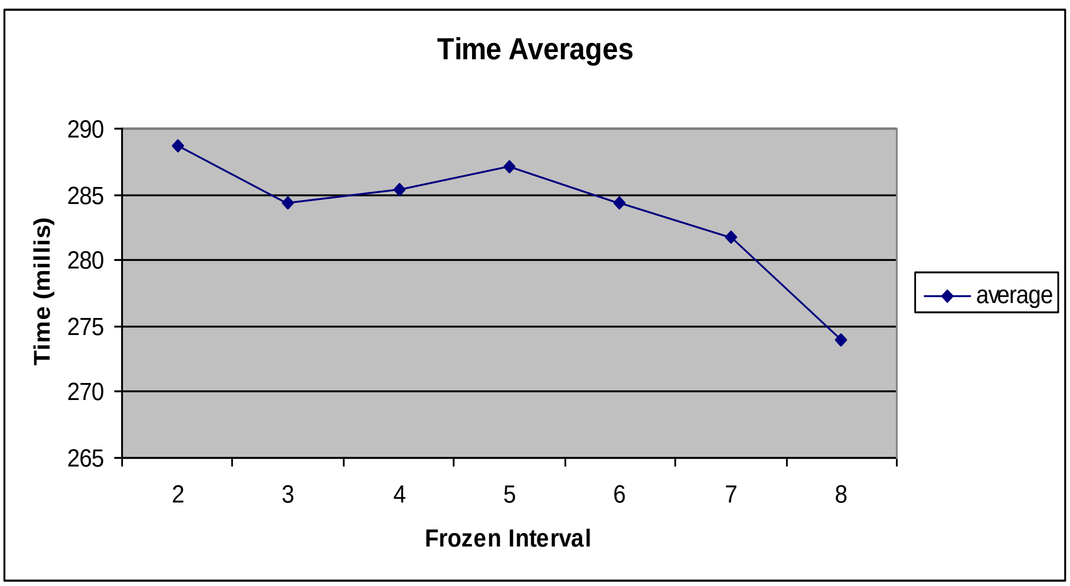


Figure17: The Time Averages for “Adaptive Reschedule” method

Figure 18 shows better chromosome sizes obtained with adaptive reschedule from that of the original algorithm that proposed by Chen.

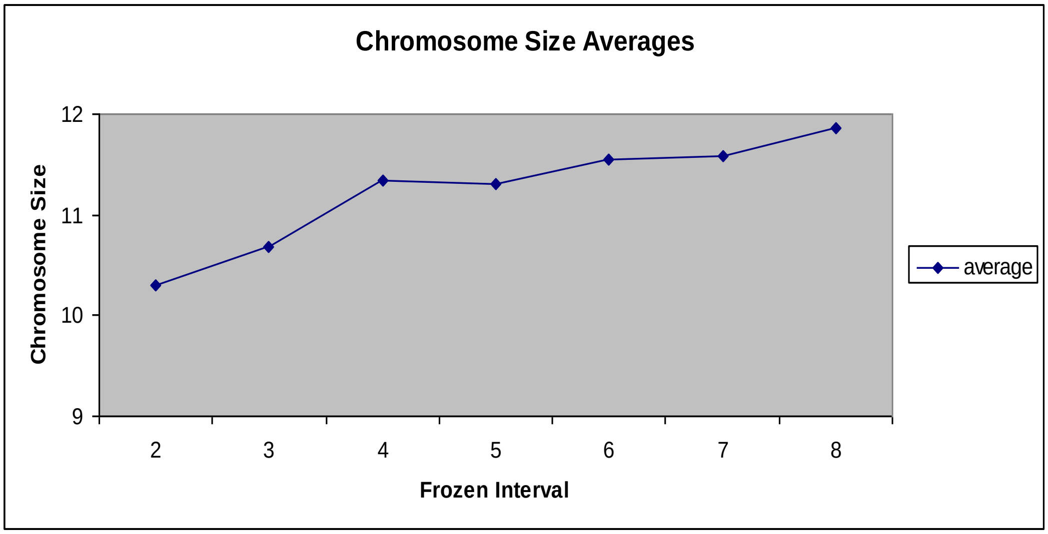


Figure18: The Averages of Chromosomes Sizes in the "Adaptive Reschedule" method

Table4 show 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.

Table 4: Percentage of enhancement gained by the "Adaptive Reschedule Interval"

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

4. Adaptive Frozen Interval:

In Figure19, 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 will get 4010 as a fitness value.

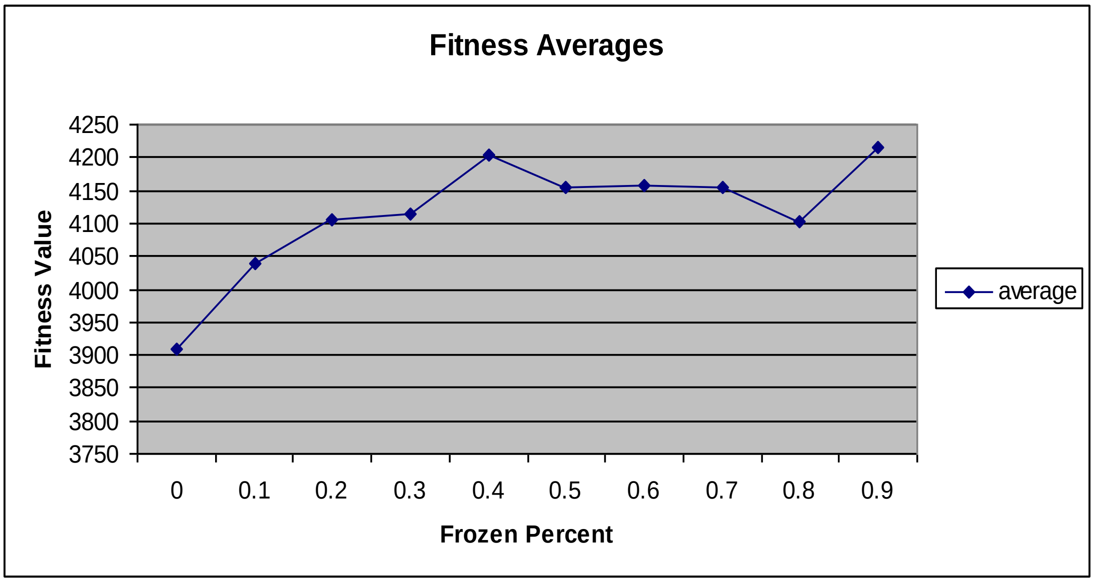


Figure19: The Averages of Fitness in the “Adaptive Frozen” method

Figure20 shows the stability averages that are obtained for all frozen interval tested percents. We notice that the stability averages are get better while frozen interval converges from the reschedule interval.

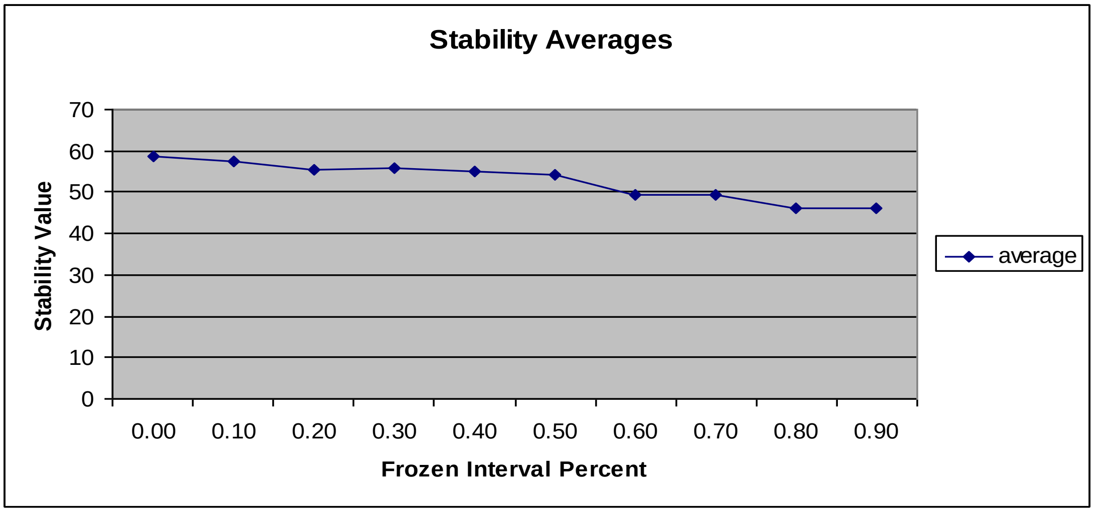


Figure20: The Averages of Stability in the “Adaptive Frozen” method

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 chose is 0.4.

5.Results Comparisons:

Figure 21 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.

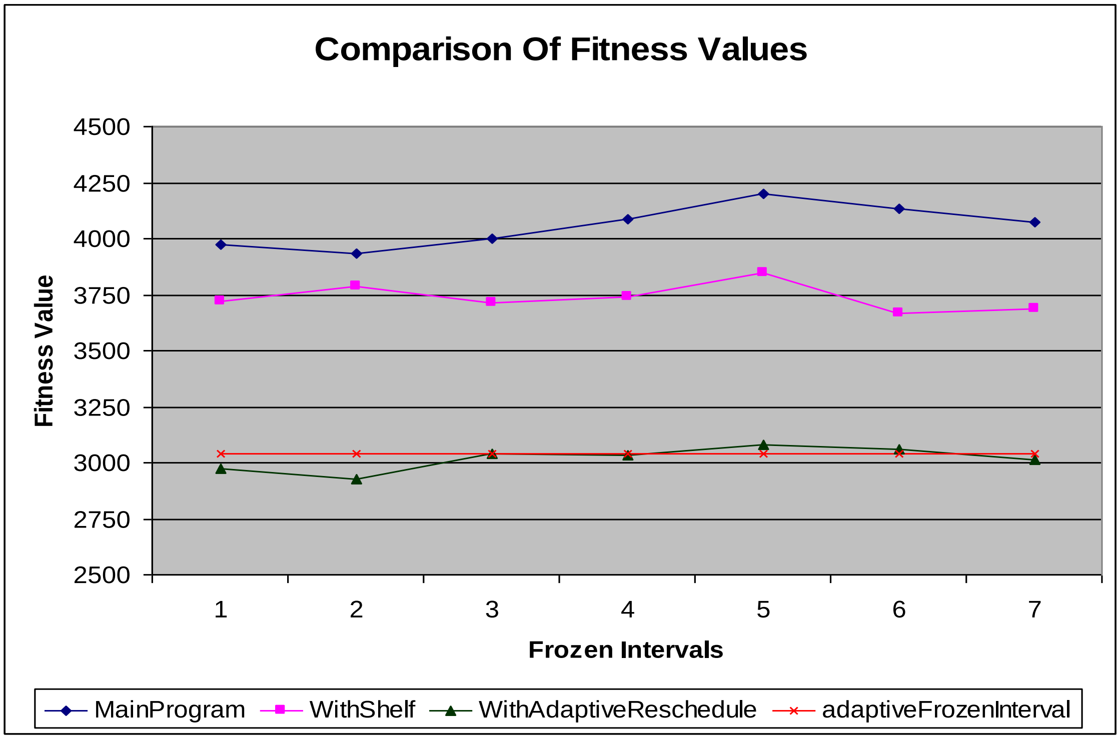


Figure21: A comparison of fitness averages obtain by the different algorithms

The Table5 below gives a clearer view about the enhancement gained by applying 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.

Table 5: Summary for the percent of fitness enhancement gained by applying each suggested algorithm to the original one

Metric \ Frozen	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

Figure 22 shows a comparison between stability values obtained when applying the different 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.

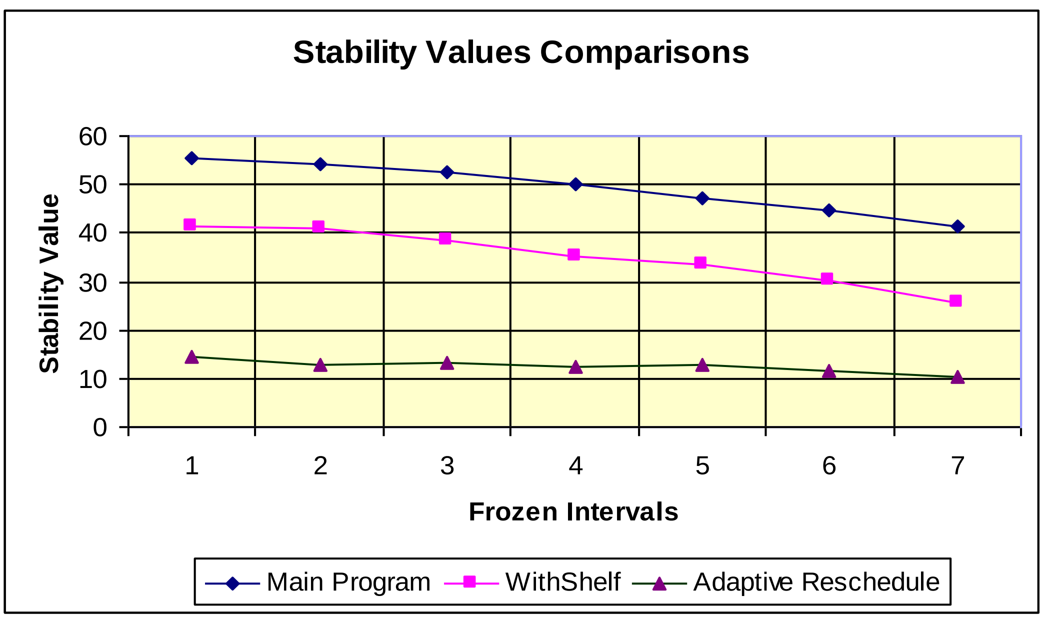


Figure22: A comparison of stability averages obtained by the different algorithms

The table 6 below gives a clearer view about the enhancement gained by applying 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 0.24 to 0.75 which is a great enhancement that affects the shop floor stability in a very dynamic production system.

Table 6: Summary for the percent of stability enhancement gained by applying the suggested algorithm on the original one

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

Table 7: Average Chromosomes Sizes Comparison

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

Figure 23 shows the comparison of the time metric between all algorithms. Chen algorithm appears by the name of MainProgram. Satisfactory results are gained.

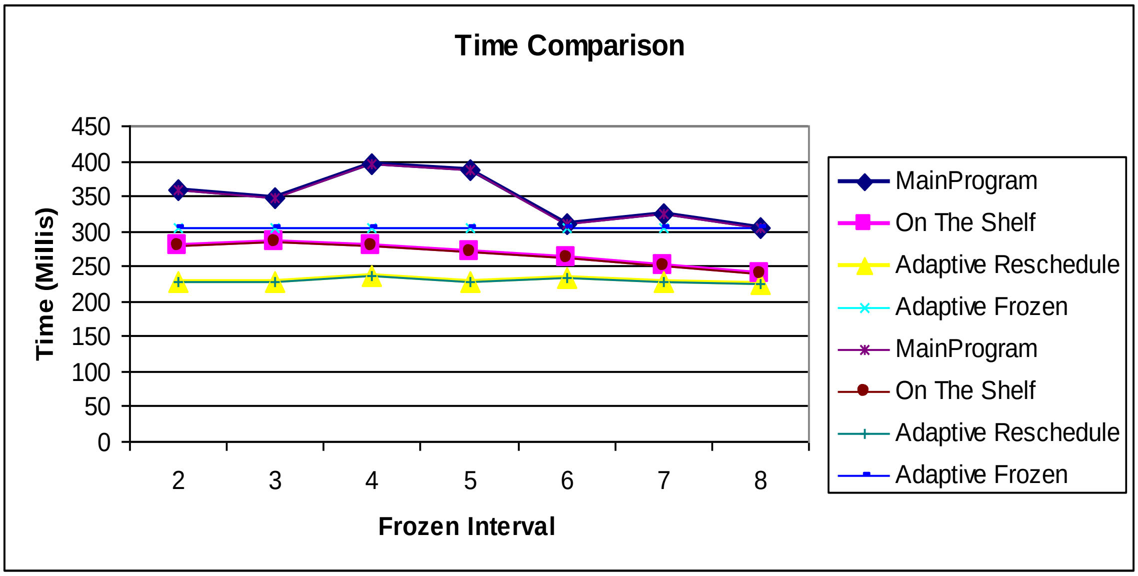


Figure 23: Time Comparison

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..

Table 8: Output Of Multi-Agent with 2 Agents

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

Better results expected to be achieved when using heterogeneous agents that defer in their 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.

CONCLUSION AND FUTURE DIRECTION

1. Conclusion:

The algorithm described by Chen 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 agent production systems.

The proposed methods are: “On the Shelf”, “Adaptive Reschedule Interval” and “Adaptive Frozen Interval”. Efficiency is enhanced by obtaining better results in terms of fitness, chromosome size, time and stability of the shop floor.

2. Future Directions

Until now no good results obtained by applying the algorithm on a MAS, but many ideas exits that 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 agent so 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 To Stock (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 these agent to submit their results.

References:

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.

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.

Bartk, R.(October 1999). 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

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.

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

Roman Barták(2002). Visopt ShopFloor. **On the Edge of Planning and Scheduling**. CP 2002, pp.587-602

Roman Barták (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.

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.

E.K. Burke, D.G. Elliman, R.F. Weare (1994).A Genetic Algorithm Based University Timetabling System, **East-West Conference on Computer Technologies in Education**, Crimea, Ukraine, 1994, pp. 35-40.

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

K. J. Chen, 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].

Cheng et al (Dec. 1998). Production Scheduling of Continuous Flow Lines. **Multiple Products With Setup Times and costs, Production and Operation Management, vol. 7, No.4**.

S. Das, 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.

Dipti Srinivasan and G. B. Tettamanzi (1995). An integrated framework for devising optimum generation schedules. **In ICEC'95 [1251]**, pp.1-4. yprog. ga95aSrinivasan.

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.

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

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

H-L Fang, P.M.Ross and D.Corne (1994). 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.

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.

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

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

Kevin R. GUE, George L. NEMHAUSER, and Mario PADRON (1997). **Production Scheduling in Almost Continuous Time, IIE Transactions 29:5**, pp. 391-398.

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

L. Han, G. Kendall, and P. Cowling (November 2002). An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem. **In Proceedings of the 4th AsiaPacific Conference on Simulated Evolution and Learning (SEAL'02)**, pp.267- 271.

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 Using Stochastic Search Methods, paper SPE 33518, Proc., 1996 NPF/SPE European 3D Reservoir Modeling Conference**, Stavanger, 16–17 April.

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

K. Kurbel, B. Schneider and K. Singh (1995). Parallelization of Hybrid Simulated Annealing and Genetic Algorithm for Short-term Production Scheduling, **in Proceedings of International Conference on Intelligent, Knowledge and Integration for Manufacturing**, Nanjing, China, March 28-31, pp321-326.

K. Poeck and C. Hestermann (1995). An intelligent interactive support system for short- and mid-term planning in production planning and control system. **In N. M. Sadeh (ed.): Intelligent Manufacturing Systems, pp.285--288. Working Notes of the Workshop held at the 14th IJCAI-95**, August 19.

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.

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, 7-10 October 2002, pp. 682--689.**

Mathijs de Weerdt and Cees Witteveen (2006). Multiagent planning: problem properties that matter. In Ed Durfee and David Musliner (Eds.). **Proceedings of the AAAI Spring Symposium on Distributed Plan and Schedule Management**, pp. 155--156.

Mathijs de Weerdt, Adriaan ter Mors and Cees Witteveen. (2005). “Multi-agent Planning: An introduction to planning and coordination”, **In Handouts of the European Agent Summer School**.

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**.

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.

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

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.

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.

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..

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

Tuson, A L (1994). 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**.

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.

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.

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.

Wiers, V.C.S., and T.W. van der Schaaf, T.W. (1997) A framework for decision support in production scheduling tasks, **Production Planning & Control, 25:2**, pp.533-544.

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.

Appendices

Appendix A

Tables of Some Numerical Results

The Main Program Fitness Values Fro 100 Run on Different Frozen Intervals

Frozen / run number	2	3	4	5	6	7	8
1	3449.297	4531.103	3316.5	3626.574	4077.561	4304.231	3040.288
2	4145.794	4050.114	4075.871	4217.835	4557.922	3427.221	4233.899
3	4072.59	4001.083	3792.831	3729.365	4149.25	4613.942	2766.427
4	5215.143	4349.545	4432.555	4826.561	5079.6	4707.793	4136.024
5	3667.518	4386.104	4176.578	4616.808	4898.203	5091.5	3792.174
6	4126.76	4047.84	3386.595	3885.807	4624.156	3461.774	5368.398
7	3387.132	4617.154	4122.433	4259.771	4389.709	3849.797	3185.938
8	5279.118	3687.198	3927.716	3739.146	3254.053	3642.852	3216.559
9	4184.106	3908.809	3784.143	3473.977	4888.678	4010.491	3504.215
10	4057.542	3564.684	4520.9	4017.661	4553.98	4995.92	4843.248
11	4148.774	3960.367	4818.589	3898.154	4116.512	4017.717	3994.357
12	4060.624	3317.745	3726.519	4368.234	4230.539	4804.359	4437.154
13	4012.617	4176.657	4063.7	3844.174	4348.096	4242.168	4214.548
14	3782.547	4135.342	3958.04	3310.979	4137.53	3772.268	3747.891
15	3449.264	2977.165	3812.72	4271.706	4128.368	3652.542	4258.098
16	2999.298	4364.852	3385.556	4330.375	4096.356	4432.022	3857.973
17	4579.628	3465.597	5315.101	3502.399	3668.171	4183.421	3381.989
18	3924.917	4880.853	3532.664	4256.39	3679.625	3911.572	4120.22
19	4630.395	3507.852	4757.766	4169.127	4508.229	3744.624	5151.872
20	3978.179	3560.95	3848.977	3812.207	3392.664	4318.711	3648.799
21	3612.392	3487.655	4001.385	4134.671	4151.807	2799.779	5269.658
22	3943.434	3358.321	3967.718	3857.118	3466.979	4733.021	3611.415
23	3849.481	2909.531	4436.311	3847.899	4222.539	4062.357	3210.393
24	4261.758	4144.453	3646.656	4500.892	4626.661	3204.89	3885.828
25	4355.203	4567.429	3516.432	4805.39	6000.233	4498.703	4752.735
26	4105.995	3873.069	4042.997	4605.041	4691.404	3709.347	3646.309
27	3909.184	3909	3799.613	4659.966	4609.266	4315.545	3633.847
28	4722.167	3615.793	4212.321	4237.648	4466.028	5605.404	3532.665
29	4958.089	3974.257	3222.808	4563.674	4041.062	4317.907	3745.955
30	3937.975	3750.014	3942.015	3796.338	4720.164	4455.941	3772.926
31	4046.65	4020.051	4767.391	4089.241	4531.041	4009.287	3802.569
32	3537.744	3574.034	3744.998	4679.213	4240.136	4945.014	4731.364

33	4181.936	4468.625	4347.021	4354.453	4041.995	4397.253	3208.353
34	4955.747	4514.707	3404.042	4457.924	4750.245	3546.406	4180.651
35	3709.3	3457.703	4433.136	4023.252	5349.022	4801.318	4192.078
36	4093.508	4317.268	3816.64	4072.347	3779.341	4430.616	4107.191
37	3864.141	4145.75	4509.293	3473.193	3995.126	3264.344	3894.387
38	4673.708	3503.834	3732.978	3838.353	3398.312	4183.96	5066.083
39	3492.315	4348.897	3642.897	4180.309	4391.599	4113.984	3181.047
40	3768.068	4956.466	4053.505	5205.915	4040.593	4131.261	4294.433
41	3928.069	4082.212	3284.559	4316.158	3760.163	3866.828	4544.89
42	3127.813	3889.375	3803.171	3978.904	4330.594	4643.286	4496.271
43	3726.982	4110.705	4583.405	4124.4	4301.896	3917.693	3582.969
44	3473.041	3563.061	4204.894	3451.179	4640.167	3615.038	3440.388
45	3565.011	4364.188	3779.055	3992.384	3635.342	3892.059	4827.488
46	4777.178	3775.313	3456.364	3331.68	4950.428	5149.841	5163.178
47	3148.014	3940.522	3909.186	3564.947	2964.837	4482.197	3186.665
48	4199.588	4036.679	3790.314	4168.966	3800.745	4525.594	3832.093
49	4074.049	4023.48	4026.277	4310.422	4099.616	4207.132	4140.313
50	4062.433	3858.171	3970.507	3857.424	4355.844	3865.282	3593.325
51	3453.4	4026.586	4854.394	3687.665	3416.184	3519.048	3406.521
52	3633.979	3814.243	3678.54	4475.456	4266.872	4015.914	4103.757
53	3511.981	3976.307	3610.858	4272.677	4131.603	4654.776	3872.465
54	3293.323	4017.755	4201.077	3368.461	3595.068	3470.652	4799.886
55	3908.828	3512.411	4152.159	3811.398	4011.957	3714.074	4992.664
56	4258.012	3682.656	3596.199	4182.462	3469.641	4245.988	4059.375
57	3650.468	4870.333	4518.063	3813.082	3656.75	4018.688	4768.554
58	3382.768	3858	3700.203	4324.713	4057.083	3200.173	3285.97
59	2802.15	3962.697	3900.823	3789.946	3724.841	4594.139	5234.61
60	3972.652	4013.151	3802.748	3585.342	5562.42	3930.136	5285.375
61	4291.588	3550.298	4139.039	3560.403	4259.595	4156.688	4297.113
62	3798.39	3470.889	4085.926	4107.092	3904.63	3748.279	3833.723
63	5092.667	2890.289	3154.8	4165.224	4664.529	4442.7	4512.675
64	3425.195	3738.237	3824.731	3490.599	4536.918	3772.267	3637.275
65	4223.083	3375.104	4645.949	3477.928	4409.163	3007.055	4134.078
66	3812.008	4095.734	3746.952	3686.718	4460.402	4174.009	4141.27
67	2705.577	3707.805	4607.688	4058.737	4395.958	4093.833	3904.578
68	4077.645	3560.068	3764.816	5104.649	3905.75	4316.743	5459.379
69	4018.121	4023.227	3391.377	4257.328	4035.778	3924.645	4334.688
70	3729.691	3667.813	4342.471	3760.42	5104.553	3895.589	3722.805
71	3965.074	4260.886	4169.682	4389.477	3560.958	4583.876	4290.253
72	3707.476	3920.972	4298.552	4139.498	3842.224	5018.969	3343.626
73	4909.25	4152.237	4937.625	3991.921	3789.375	4621.839	3334.889
74	3376.414	4475.835	3383.518	4556.028	5007.361	2974.048	3952.834

75	3230.761	3709.481	4468.085	3808.801	3959.846	4704.393	3668.292
76	3506.952	3857.856	4754.011	4942.905	5000.662	4414.733	4338.261
77	5115	3506.498	4744.264	4108.831	4322.449	4392.432	3236.666
78	4964.012	3869.342	3732.848	4597.736	4028.22	3486.544	3885.751
79	4008.003	3996.525	3473.111	4552.557	3726.543	3978.264	4744.849
80	4286.26	3643.5	3825.546	4553.666	4718.629	4166.286	4142.078
81	3689.783	4256.5	4062.73	4827.719	4491.852	4495.689	3572.937
82	4153.41	4372	3634.105	3595.939	3671.418	4146.116	4321.693
83	4259.575	3749.146	3486.516	4720.927	4641.689	5295.813	3573.617
84	4016.656	3799.534	3815.691	3798.252	3869.379	3582.315	4134.054
85	3692.386	4820.629	3957.132	4206.759	3233.449	4811.049	4076.946
86	3930.453	3992.765	3293.956	3867.361	3589.441	3247.102	3613.059
87	4005.09	4252.705	4666.708	3833.733	4759.509	4858.098	4561.351
88	3442.357	4006.812	3347.397	4404.942	3074.671	4582.642	4049.549
89	3857.04	3369.172	3536.556	4100.222	4365.029	5374.007	4104.471
90	4513.453	3803.386	4246.361	3534.682	3022.485	3343.496	4882.493
91	3989.358	3270.301	4631	3426.038	3903.992	4532.763	4450.566
92	3912.992	4740.703	4199.375	3942.345	4238.952	3451.656	4174.248
93	3911.178	3769.21	4127.951	4089.91	3995.709	3749.604	2890.14
94	4568.647	3358.722	4795.921	4629.741	5095.133	4048.924	4863.787
95	3558.092	3521.87	3795.451	3900.771	4591.074	3815.104	4962.111
96	4320.182	3772.631	4467.199	4381.476	4634.875	2899.923	3558.682
97	3595.703	4297.977	3847.766	4428.24	4278.393	4225.2	4435.031
98	4694.076	4613.995	4418.005	3291.041	4450.712	3241.051	4621.105
99	3659.691	4149.547	3586.126	4641.904	3463.177	4730.898	3999.187
100	3992.899	3902.477	4028.235	3817.663	3838.635	5092.259	4203.919
average	3970.86	3930.584	4002.539	4086.979	4198.679	4136.367	4072.007

The Main Program Stability Values Fro 100 Run on Different Frozen Intervals

run number \ frozen	2	3	4	5	6	7	8
1	54	63	53	53	42	28	34
2	51	58	39	54	49	38	29
3	59	66	55	54	47	43	37
4	74	76	61	45	57	43	61
5	61	53	39	56	35	53	40
6	44	39	56	52	37	39	36
7	69	49	40	44	57	27	41
8	46	66	57	47	39	47	42
9	49	52	49	32	61	50	39
10	53	43	50	47	41	53	44
11	73	43	57	61	47	32	48
12	57	58	47	37	51	49	60
13	58	41	41	57	48	46	29
14	51	53	51	46	39	57	32
15	49	48	40	44	57	31	36
16	69	57	52	59	45	46	47
17	56	51	67	54	59	63	37
18	45	71	46	49	50	50	49
19	55	48	67	51	46	55	29
20	53	52	46	44	41	65	57
21	47	50	53	48	55	53	38
22	49	80	40	38	58	63	28
23	52	43	40	61	43	47	33
24	42	51	53	54	49	32	36
25	61	61	51	49	40	49	54
26	48	73	51	47	63	32	36
27	44	51	61	35	55	33	39
28	61	41	41	56	59	49	47
29	55	42	46	66	56	38	39
30	51	52	44	49	34	66	61
31	47	71	71	44	44	46	34
32	64	57	48	52	44	47	59
33	59	54	59	54	57	36	53
34	50	47	64	54	32	38	42
35	55	52	62	41	71	41	56
36	50	44	47	44	40	67	58
37	64	61	53	45	40	55	39

38	60	57	43	49	56	46	39
39	53	60	65	69	48	47	33
40	51	74	39	51	47	50	49
41	73	58	41	58	38	39	47
42	59	61	53	45	40	43	52
43	64	45	45	43	56	51	36
44	52	47	44	45	47	41	38
45	51	54	46	46	52	37	44
46	68	47	47	46	61	40	35
47	55	52	60	34	43	45	21
48	43	73	65	36	55	53	46
49	63	61	57	58	28	46	31
50	50	51	39	49	48	54	51
51	57	42	60	41	42	31	55
52	52	42	47	54	42	36	51
53	38	48	42	41	32	48	31
54	49	59	65	56	31	35	44
55	47	48	60	43	36	43	33
56	60	50	50	46	36	30	29
57	47	87	55	51	31	51	49
58	58	52	62	61	33	38	26
59	58	48	77	48	55	48	35
60	70	46	48	47	59	38	37
61	45	63	74	48	44	68	43
62	42	51	42	55	45	40	34
63	60	49	56	49	41	48	37
64	65	38	46	52	44	41	37
65	55	69	72	38	46	52	47
66	70	42	57	55	47	47	32
67	55	58	53	56	50	41	43
68	46	57	47	52	40	31	69
69	45	44	45	55	49	37	34
70	56	54	42	42	55	31	28
71	41	40	44	59	59	41	50
72	50	41	66	43	33	62	43
73	71	63	57	53	39	40	30
74	48	51	56	56	45	39	28
75	54	53	63	43	46	56	52
76	46	52	52	49	56	48	34
77	64	38	61	56	42	45	42
78	44	70	39	65	35	52	28

79	65	65	58	48	33	42	37
80	71	49	40	40	41	40	54
81	46	54	51	69	48	47	37
82	59	55	69	61	60	40	44
83	79	52	52	51	36	56	35
84	62	59	58	56	51	42	46
85	70	54	39	65	42	38	32
86	72	46	58	54	42	29	55
87	68	76	53	45	57	43	43
88	52	44	46	51	59	44	34
89	40	55	53	53	42	60	42
90	64	42	62	49	46	37	48
91	45	50	52	39	43	35	44
92	56	68	53	52	52	37	38
93	46	49	69	46	70	62	43
94	48	53	51	49	46	35	45
95	50	53	47	50	52	45	44
96	84	76	57	63	61	34	49
97	43	55	43	44	49	44	49
98	57	64	39	52	55	39	27
99	66	40	64	58	61	39	28
100	56	53	45	53	57	54	47
average	55.59	54.24	52.38	50.14	47.23	44.58	41.23

The "On the Shelf" method Fitness Values for 100 Run on Different Frozen Intervals

Frozen / run number	2	3	4	5	6	7	8
1	3850.794	3703.368	3428.588	3527.943	3252.039	3753.291	4517.017
2	4290.154	4345.77	3930.061	4108.935	4259.382	3797.396	3960.464
3	3841.721	3689.684	3387.586	3865.584	3382.156	3500.094	5019.219
4	3093.931	3024.632	3798.092	3698.542	3429.395	4586.041	2994.78
5	3622.093	3477.343	3379.194	4232.041	4038.046	3415.469	4125.987
6	4066.089	3713.356	3155.007	3349.463	4099.056	3120.158	4003.854
7	3878.08	3555.786	3828.879	2966.084	4234.893	3411.779	3944.375
8	3749.391	3653.885	3870.586	3463.168	4127.013	3822.648	4385.632
9	2860.405	3482.622	4217.094	3860.063	4312.055	3391.651	4434.698
10	3355.179	4220.668	4151.893	3644.519	4324.649	2944.185	3101.673
11	3659.658	3627.287	3204.976	3821.468	3283.628	4221.391	3576.047
12	3368.014	3789.926	3697.723	3616.034	3788.832	3932.974	4378.685
13	3307.763	3640.439	3571.439	3510.781	3380.987	2790.585	2732.627
14	3248.632	3498.146	2977.609	3102.526	4204.077	4240.804	3533.884
15	2829.645	3540.419	4101.922	3404.604	3912.796	3920.766	3552.545
16	3467.148	4039.838	3373.965	3681.681	3178.601	3367.893	4614.588
17	3270.517	4062.563	3209.091	4561.257	3465.204	3954.409	4096.318
18	4545.934	3292.976	3678.511	3365.183	3902.209	2978.998	4865.242
19	4091.646	4005.451	3706.071	4009.794	4279.027	4227.472	3733.169
20	3962.035	3962.985	4585.998	3744.995	2513.376	2232.728	3344.313
21	3743.236	3288.079	3303.87	3606.404	4957.422	2778.723	4599.202
22	5026.664	4531.188	3577.382	4050.271	4001.421	2627.89	2660.082
23	3304.721	3684.458	3382.102	3329.029	4425.653	4953.448	3679.086
24	3294.915	3517.033	4181.729	3240.161	3484.987	4104.62	2800.717
25	3699.638	3143.743	4045.375	4070.487	3235.396	3207.26	3244.292
26	3646.363	4235.141	4200.707	3947.602	3727.557	4001.605	4449.257
27	3654.399	3657.834	3600.269	4141.667	4199.346	3187.641	3062.513
28	3253.789	3627.223	3244.132	3462.836	3351.888	3189.574	4018.438
29	4344.248	4333.663	4007.553	4064.354	4506.961	4097.38	3377.211
30	3138.967	3946.144	3451.191	3718.35	3899.94	3946.271	2984.583
31	2831.597	3593.601	4253.175	2843.911	4716.68	4501.547	3226.117
32	3874.114	4870.067	4020.101	3725.568	4347.454	3928.75	3827.542
33	3932.379	3829.544	3669.461	4285.19	3097.683	3251.488	3065.097
34	3952.09	3880.788	3482.591	3499.403	3482.471	3244.935	4814.314
35	4045.136	3565.836	3327.597	4144.285	4170.476	3154.171	4533.14
36	3768.379	4432.398	3709.611	3980.266	3790.328	3888.004	3343.744

37	4042.284	4389.739	3463.75	2969.801	3878.967	4102.022	4216.538
38	2747.921	3218.917	3149.328	3454.477	4487.414	3451.323	3513.102
39	4015.563	3811.091	3834.431	3665.313	4397.891	3470.022	3558.485
40	4233.276	3676.954	3762.202	3557.258	3782.023	4172.366	3805.797
41	3832.28	3119.455	3694.94	3251.177	4125.471	3416.019	3551.36
42	3551.236	3342.056	4351.27	3919.143	4072.588	4267.191	4199.802
43	3493.681	3366.452	4766.719	4161.641	3494.146	3636.414	3484.317
44	4181.139	3498.568	3681.591	3533.058	3840.768	3023.316	3706.993
45	4226.045	3977.656	3450.646	4254.015	4029.176	3623.311	3303.204
46	3487.938	2447.419	4026.439	4175.451	3148.77	3751.338	4402.047
47	3512.002	3084.526	3907.694	3401.902	3492.164	3848.985	3854.162
48	4297.969	4191.814	4047.245	3394.691	3737.985	3152.162	3066.847
49	3185.752	3334.674	3677.081	3034.347	3698.455	4105.675	3918.281
50	2920.334	3493.16	4371.568	3548.947	3382.554	3215.309	3136.123
51	3652.967	2991.101	4226.75	3975.317	4273.305	4016.149	5057.527
52	3261.25	3279.299	4921.764	3443.729	4543.008	4450.313	4533.369
53	4401.368	4047.193	3786.197	3586.551	3887.902	3143.338	2842.279
54	3215.481	3850.771	3622.84	4296.036	3933.657	3595.074	3876.518
55	4419.927	4722.564	3837.899	3884.44	3123.685	3402.097	2948.69
56	3612.686	3330.455	3423.661	3360.17	4231.667	2920.945	2914.186
57	4018.379	4771.431	3246.743	3736.827	4277.391	3285.434	4271.483
58	3816.847	3940.826	3447.619	4185.093	2953.669	4149.446	3546.687
59	3590.571	3313.786	3803.48	3223.392	3776.432	2892.335	2935.636
60	4053.092	3860.561	3768.26	3825.213	4335.531	3498.238	3314.243
61	4103.359	3659.272	3348.167	4041.145	4610.712	3189.382	3258.818
62	3619.293	3857.956	3547.095	3992.199	4530.818	3061.514	3192.421
63	3454.542	3643.982	3675.563	3510.517	2561.166	3300.373	2252.851
64	3101.882	4283.029	4166.153	4302.383	3697.122	3135.645	3528.459
65	3839.537	3465.761	4036.25	3562.809	3342.06	4404.648	3703.951
66	3613.022	4241.365	3517.885	3921.583	3717.967	4478.824	3708.211
67	3605.059	3904.74	4738.639	4737.742	3995.26	3921.25	3510.686
68	3957.233	3973.288	3595.356	3528.794	3600.416	4593.51	3065.967
69	3643.865	4411.547	3962.515	4016.428	3986.671	3608.5	3402.377
70	3507.103	4615.094	3326.929	3978.008	4544.898	4082.753	3730.069
71	3984.352	4179.062	3403.322	3881.662	3823.104	3889.886	3574.468
72	3519.035	3388.603	3690.419	4205.729	4598.029	3933.014	3716.056
73	3526.062	3690.272	3606.682	4057.5	4123.059	3949.02	2673.57
74	3879.199	4312.43	4156.891	3982.492	3653.582	4259.639	2868.579
75	3009.205	3989.085	2602.739	3745.646	3667.738	3775.12	3211.737
76	3793.297	3666.404	3414.479	3590.353	3975.711	3863.424	3913.371
77	3906.077	3629.419	4724.303	3482.242	4446.046	4195.101	3964.253
78	3831.454	4236.786	3081.375	4283.164	3915.91	3378.231	4294.307

79	3762.518	3372.1	3239.526	3443.571	4232.792	3981.649	3565.292
80	3490.056	3596.71	3098.851	3447.681	3920.653	3259.02	4143.361
81	4156.522	3617.469	3998.525	3906.101	3732.079	3998.736	2861.401
82	2625.062	3876.043	3766.816	3913.441	3938.523	3985.401	3511.547
83	4243.35	4071.682	4012.891	4051.653	4131.264	3405.617	3502.79
84	5136.914	4279.515	3094.864	4241.304	3815.726	4481.189	3188.791
85	2374.001	2819.571	3261.147	4106.943	3915.263	4099.282	4778.016
86	4497.413	4459.229	4076.764	3315.793	3602.537	4171.099	4444.222
87	3747.838	3693.987	3799.535	4041.304	3188.669	3897.349	4398.351
88	4260.722	2930.323	3878.403	3132.36	4143.944	3419.94	4606.696
89	3052.462	4249.514	3875.141	3368.503	3530.884	3848.523	4070.247
90	4445.556	3882.248	3236.124	3343.188	3793.968	4349.949	4024.157
91	3957.222	4058.057	3561.742	3698.434	3817.731	4060.611	2725.575
92	3505.697	4067.379	3576.059	3967.836	4361.977	3244.188	3530.965
93	3683.055	3975.222	3937.871	3572.374	3064.455	2886.886	3469.029
94	3331.429	3960.014	4050.016	3454.463	4198.858	3465.586	4120.141
95	3717.101	3793.922	4391.464	3420.722	3646.345	3141.882	3104.953
96	3704.712	3602.096	3747.55	4158.54	3748.309	3549.409	4073.272
97	3758.954	3555.296	3680.948	3404.75	3695.828	2291.333	3282.273
98	4578.062	3875.866	3606.015	4401.974	3381.175	4284.39	3311.966
99	4377.963	4723.758	2598.23	3300.517	3574.537	3587.146	3052.331
100	3758.712	3743.699	3415.311	4090.646	2587.265	3989.721	4878.695
average	3716.983	3788.761	3714.798	3740.849	3844.808	3666.816	3687.723

The "On the Shelf" method Stability Values for 100 Run on Different Frozen Intervals

run number \ frozen	2	3	4	5	6	7	8
1	34	38	32	30	42	28	37
2	39	50	35	29	29	30	37
3	32	22	43	52	34	63	19
4	35	29	30	27	33	49	17
5	32	47	28	44	40	32	34
6	44	34	29	43	34	24	27
7	44	43	35	46	28	24	17
8	38	47	50	35	35	44	31
9	31	27	35	35	30	27	31
10	50	41	32	36	49	40	19
11	51	21	50	38	33	28	23
12	57	44	48	29	40	33	25
13	39	36	32	40	24	40	20
14	42	27	35	42	21	36	16
15	42	29	37	38	42	28	21
16	43	36	43	29	22	23	27
17	55	36	46	33	32	24	22
18	42	51	60	30	30	21	50
19	45	45	45	41	32	36	14
20	40	37	38	37	47	25	23
21	49	41	39	39	29	40	26
22	48	55	31	26	39	38	35
23	46	39	34	38	39	22	17
24	36	38	34	26	43	30	20
25	38	28	32	33	31	19	21
26	46	36	49	43	30	28	29
27	52	43	34	43	31	33	22
28	50	45	51	36	30	30	29
29	45	44	37	53	32	38	22
30	39	51	39	30	26	33	21
31	58	31	38	32	35	39	26
32	53	39	37	30	16	27	35
33	50	39	40	36	39	33	30
34	38	60	37	31	33	20	33
35	48	37	42	47	40	30	25
36	43	38	32	21	22	21	28

37	36	58	31	30	37	21	33
38	44	34	35	29	29	20	26
39	36	55	31	32	40	30	24
40	42	50	33	47	33	36	21
41	30	39	26	48	32	33	29
42	47	31	31	21	36	32	21
43	41	37	40	33	24	17	29
44	42	40	35	24	28	30	23
45	43	40	33	28	33	38	18
46	39	41	43	39	33	27	33
47	46	38	53	42	34	41	27
48	37	36	40	27	33	25	17
49	54	32	44	39	44	31	44
50	52	39	41	35	35	25	21
51	45	38	28	43	46	34	21
52	35	37	58	29	46	27	26
53	38	33	33	41	29	32	27
54	37	37	25	36	47	43	29
55	56	38	47	25	41	36	21
56	47	33	34	40	32	30	27
57	40	38	49	39	23	34	24
58	38	52	29	34	33	28	30
59	44	41	46	23	30	29	20
60	45	52	40	36	45	32	36
61	42	51	35	24	42	38	33
62	30	48	36	28	39	20	19
63	32	46	42	28	28	21	26
64	35	43	36	37	27	37	34
65	27	34	27	31	28	30	28
66	34	43	41	39	30	22	26
67	28	38	32	38	33	31	33
68	27	58	27	51	47	29	19
69	37	37	42	36	31	30	18
70	51	49	31	31	23	31	21
71	36	48	34	46	43	24	21
72	32	48	37	39	24	26	32
73	33	35	39	36	27	15	29
74	41	48	36	22	38	34	16
75	37	49	59	30	31	25	21
76	29	42	38	36	32	25	41
77	39	37	41	41	40	24	35

78	36	56	47	35	32	31	23
79	41	33	33	36	30	42	17
80	36	31	36	33	35	24	26
81	43	36	48	38	33	35	20
82	41	47	38	41	23	38	22
83	43	40	39	35	27	26	26
84	56	32	39	30	30	37	17
85	37	43	51	47	43	24	28
86	37	34	51	33	30	36	28
87	53	33	26	35	22	26	19
88	46	34	39	32	26	26	25
89	39	50	44	34	52	22	24
90	32	46	51	31	35	29	30
91	45	49	27	37	31	35	37
92	45	46	45	38	27	22	46
93	39	38	37	42	46	24	27
94	31	47	34	36	29	30	25
95	36	40	37	41	35	31	21
96	50	49	31	35	27	38	22
97	34	42	46	30	27	24	21
98	45	44	40	26	34	30	20
99	47	41	38	31	41	30	19
100	44	47	39	25	38	29	25
average	41.34	40.85	38.43	35.12	33.51	30.18	25.76

The "Adaptive Reschedule" method Fitness Values for 100 Run on Different Frozen Intervals

Frozen / run number	2	3	4	5	6	7	8
1	2638.221	3068.464	3051.814	3414.75	2819.206	2496.968	3010.818
2	3478.179	2836.774	2761.298	2896.828	3423.321	2305.688	2882.735
3	3068.818	2476.544	2893.078	3049.328	2837.559	2819.206	2852.27
4	3347.036	2713.231	3030.304	3224.896	3444.179	3341.036	3195.161
5	2889.73	3478.179	3232.872	2443.586	2981.727	3089	3454.179
6	2732.122	3054.667	3343.229	3158.007	2852.122	2999.545	3290.091
7	2975.518	2951.351	2850.039	3149.088	2786.5	3228.924	2998.147
8	2890.954	2938.378	3401.893	3445.607	3294.063	3364.063	3858.179
9	2974.638	2776.406	2893.078	3007.853	3254.063	2949.441	2943.182
10	3017.27	3219.063	3294.063	2893.078	3237.396	3307.601	3477.607
11	2787.692	2877.474	2686.426	3431.893	3311.563	2935.182	2801.727
12	2767.731	3147.466	2533.809	2841.794	2826.971	2837.559	2301.938
13	3361.5	2716.332	3478.179	3093.073	3091.699	3289.896	2894.029
14	3669.172	2875.676	3355.607	2979.794	3231.402	2846.735	2950.455
15	2959.135	2939.112	3020.667	3185.691	2946.818	3136.115	2496.968
16	2840.828	2899.904	2535.529	3432.75	2793.059	3293.229	3003.703
17	2785.016	2835.288	3196.385	2966.971	2837.559	2830.5	2893.078
18	2784.632	2535.529	3347.036	2894.906	3503.321	3299.063	3058.632
19	2958.486	2630.426	3084.74	2983.324	3335.174	2935.182	3359.909
20	2987.333	2564.016	3260.98	2935.182	3236.926	3452.147	2893.078
21	3041.563	3315.729	2662.067	3377.036	3044.901	3500.464	2848.853
22	3405.321	3355.729	3336.007	3041.563	2893.078	3401.893	2893.078
23	2878.105	2893.078	2981.727	2981.727	3147.466	3232.061	2893.078
24	2787.692	3227.396	2936.946	3364.063	3153.142	2894.029	3098.073
25	3329.063	2786.5	2978	2879.206	3143.412	2892.273	2878.853
26	2787.789	2716.897	2970.929	3330.174	3423.321	2965.456	2807.122
27	2859.476	2819.206	2952.24	2981.727	3272.736	3326.284	2830.5
28	2877.474	2996.757	3153.142	3467.674	3415.036	2470.5	3230.697
29	2986.116	2961.73	3298.507	2826.859	3415.036	2893.078	2879.206
30	2658.221	2742.067	2633.956	2834.029	2496.968	2468.779	3359.909
31	3195.161	3144.223	3401.893	3303.229	2831.098	3469.607	3230.697
32	2731.692	3225.304	2925.061	3216.563	3320.729	2835.545	2896.828
33	2877.474	3099.628	2788.698	2785.324	3401.893	3364.063	2301.938
34	2893.078	2524.266	3057.533	3431.893	2894.906	2981.727	2301.938
35	2870.27	3297.396	3022.27	3014.901	3240.729	2875.676	2305.688

36	2904.632	2868.395	3530.75	3364.063	2877	2981.727	3380.439
37	3356.672	3052.007	3140.169	2916.233	3063.454	2893.078	3544.607
38	3411	2595.132	3328.229	3324.896	3482.464	3292.872	2799.955
39	2861.053	2661.069	3301.563	2923.545	2822.735	2950.455	3500.464
40	2877.474	2619.955	2845.263	2953.229	3020.16	2893.078	2470.5
41	2926.295	2780.923	3269.493	3227.396	2879.206	3580.464	2668.174
42	3319.063	3171.75	2630.426	2524.544	3239.063	2496.968	3102.455
43	2935.164	2734.375	3245.845	2949.476	2788.853	3236.563	2935.182
44	2901.474	2788.853	3368.229	2609.135	3007.853	3516.179	2985.364
45	2838.701	2855.428	2670.529	2913.229	3183.953	3511.893	3102.455
46	2988.729	2875.676	3447.607	2595.132	3401.84	3478.179	2945.422
47	2793.846	2754.03	3183.412	3409.607	3368.229	3500.464	2935.182
48	2468.779	3132.061	3254.063	2732.75	3412.813	2721.794	2468.779
49	2954.089	2907.789	2877	2784.227	3149.899	2991.217	3137.036
50	3350.25	3152.914	2981.727	2830.5	2831.182	2893.078	3100.636
51	2950.455	2721.794	2879.206	2893.078	3175.304	3244.896	2935.182
52	2734.769	2433.786	2875.676	3449.893	3139.896	3364.063	2774.375
53	3297.396	2934.203	3580.464	2613.516	2792.382	3447.321	2867.559
54	2953.333	2605.144	2845.97	3228.007	3473.893	3269.493	3445.607
55	2939.027	3058.667	3359.896	3239.899	3009.638	2893.078	3328.229
56	3115.845	2938.818	3294.063	2875.676	3228.924	2796.385	2733.088
57	2968.865	3323.229	2468.779	2837.559	2886.385	2776.406	3542.036
58	2719.385	2714.808	2916.233	3080.169	2733.088	2301.938	2826.265
59	2786.5	3196.8	2893.078	2896.828	2786.5	3463.441	3218.277
60	2724.227	2877	2796.859	2938.818	2875.676	3087.533	2956.179
61	3200.729	2788.308	3433.607	2851.332	2819.206	3121.48	3228.924
62	2941.338	3065.333	2859.476	2935.729	2897.122	2825.912	2841.794
63	3054.667	3089.088	2671.442	2780.28	2977.607	3452.147	2851.368
64	2780.923	2965.622	2630.426	3007.453	3028.277	2856.217	3213.161
65	3299.25	3401.893	2788.273	3445.607	2893.078	2894.029	3359.909
66	2928.667	2893.078	2786.5	2686.426	2496.968	2823.182	3149.899
67	2993.721	2875.676	2830.5	3049.328	3293.507	3013.237	2301.938
68	2844.071	3478.179	3049.328	2771.118	3259.896	3127.196	3248.618
69	2776.332	2786.5	3278.464	2923.545	3299.493	3368.229	4128.464
70	3186.19	2955.243	3427.607	2965.729	2822.735	3269.358	2826.265
71	3248.46	3082.601	2443.586	3415.036	2882.735	2923.545	2799.955
72	2856	2751.789	2888.636	3140.169	3443.321	2893.078	3104.273
73	2941.622	3094	3249.063	2893.237	3223.953	3289.896	3228.924
74	2898.365	2880.561	3188.007	3164.896	2470.5	2832.533	2988.395
75	3496.547	2742.191	2670.529	3193.958	3511.179	2624.519	3062.379
76	2764.923	3054.667	3423.321	2595.132	2661.298	2470.5	3271.206

77	2951.743	2965.071	3227.396	2923.545	3319.896	3042.455	3265.304
78	3292.179	2916.406	2949.076	2786.5	3348.507	3402.179	2496.968
79	2806.737	2845.324	2872.449	2898.656	3187.396	2979.794	4041.036
80	3264.75	2871.158	3149.088	3441.321	2678.586	2898.656	2960.464
81	2721.794	2524.544	2772.533	2946.818	2893.078	3483.321	2468.779
82	2910.947	2786.5	2893.078	3199.063	2721.794	2693.606	2837.559
83	2720	3040.573	3333.321	2893.078	2935.182	2967.161	2935.182
84	3414.75	2896.216	3135.304	3445.607	2923.545	2830.5	3999.607
85	2788.853	3062	3149.088	3155.036	3228.924	3285.729	2524.544
86	2874.316	3097.333	2786.5	3183.953	2950.439	3084.74	2893.078
87	3116.162	3437.893	2882.406	3319.896	2841.794	2301.938	2312.906
88	2801.053	3093.073	2981.727	2819.206	3441.321	2981.727	3230.697
89	2814.154	2893.078	3150.75	3293.229	3414.75	3086.655	2953.956
90	3071.392	2748	2595.132	2786.5	2822.735	2793.059	2896.828
91	2877	2865.364	3298.229	2889.227	4161.463	3091.48	2852.382
92	2835.129	2801.231	2988.395	3045.954	3223.953	3441.321	3427.607
93	2950.164	2893.773	3228.75	2954.089	2790.136	2788.853	3931.036
94	3184.157	2826.818	3324.896	2893.078	3324.063	3431.893	2305.688
95	2846.595	2857.044	3048.906	2533.809	3324.34	2748.701	4141.321
96	2966.743	3090.667	2977.607	2979.794	2667.452	3554.893	2878.853
97	2941.368	2776.332	3431.893	3294.063	3208.618	3085	3316.382
98	2830.5	3098.057	2684.375	3252.061	2786.5	3419.036	2500.839
99	2935.164	3252.396	2830.5	3137.036	3422.464	3332.464	2623.591
100	3365.036	2961.73	3183.953	3188.007	2935.182	3272.736	2894.029
average	2976.62	2929.001	3038.286	3034.543	3083.025	3056.94	3013.699

The "Adaptive Reschedule" method Stability Values for 100 Run on Different Frozen Intervals

run number / frozen	2	3	4	5	6	7	8
1	21	17	20	9	12	9	13
2	15	15	27	8	16	9	8
3	14	15	3	8	7	5	28
4	8	23	22	11	14	14	15
5	11	12	19	13	6	6	5
6	23	6	13	12	17	9	9
7	21	13	14	14	11	4	11
8	19	13	7	7	10	15	10
9	13	27	2	20	9	10	7
10	10	13	12	8	9	12	22
11	11	14	17	7	12	5	11
12	14	21	12	6	9	6	6
13	23	15	12	14	19	8	3
14	19	9	9	9	14	4	3
15	10	16	13	15	3	17	6
16	14	14	11	14	25	18	7
17	20	21	18	9	5	9	6
18	18	12	9	1	22	8	9
19	10	7	15	10	17	3	9
20	9	26	13	8	22	8	2
21	16	14	23	19	17	6	10
22	17	15	16	19	8	7	9
23	12	3	8	2	13	19	4
24	12	12	22	12	11	6	15
25	19	15	13	3	15	4	11
26	19	15	16	23	14	11	24
27	19	10	15	8	14	14	5
28	14	6	13	20	15	19	3
29	25	8	13	18	15	4	9
30	19	21	8	6	10	18	6
31	10	13	6	7	26	10	5
32	20	23	11	13	13	5	1
33	11	13	28	11	7	14	11
34	2	25	14	11	3	5	6
35	15	13	19	17	11	7	12

36	9	14	13	14	7	2	13
37	17	10	15	15	23	5	16
38	14	9	15	8	4	18	13
39	11	26	18	6	8	3	5
40	14	11	16	22	20	5	17
41	21	17	17	13	5	13	27
42	18	19	11	13	17	10	11
43	14	28	16	18	11	11	4
44	13	17	13	34	21	12	2
45	20	19	28	19	10	13	8
46	11	6	20	9	16	9	17
47	11	24	19	12	10	4	6
48	15	12	10	12	15	14	17
49	19	9	8	18	12	16	17
50	15	5	4	10	5	8	6
51	8	10	7	1	18	10	4
52	23	18	7	5	14	9	28
53	16	14	11	32	9	7	11
54	13	20	9	17	8	19	11
55	12	7	14	17	20	1	12
56	7	5	8	7	4	25	11
57	9	9	19	7	22	27	14
58	21	20	12	18	9	8	10
59	8	3	5	2	13	7	24
60	21	7	19	4	5	12	20
61	18	17	10	20	7	20	3
62	12	4	16	17	21	16	6
63	9	20	20	18	20	8	21
64	13	9	7	6	20	21	15
65	16	6	7	7	3	7	10
66	14	5	11	14	7	5	15
67	21	7	10	8	16	20	8
68	16	8	11	11	9	13	5
69	22	11	10	6	14	13	14
70	23	9	11	22	7	19	9
71	25	13	15	16	4	4	12
72	12	23	5	14	17	3	6
73	13	5	12	32	20	9	4
74	15	9	15	17	17	13	20
75	18	13	24	13	11	33	11
76	14	6	11	9	33	18	4
77	15	16	15	3	7	7	17

78	11	14	9	7	19	19	6
79	18	6	20	6	13	15	15
80	11	8	9	11	26	3	18
81	10	13	25	5	1	9	18
82	11	11	4	13	14	23	6
83	17	18	8	2	4	18	5
84	13	14	24	10	4	8	7
85	10	5	18	17	4	10	15
86	10	6	7	12	17	18	13
87	5	10	25	13	12	12	6
88	21	14	2	8	14	2	3
89	16	9	15	14	11	19	18
90	10	18	9	14	8	23	3
91	10	8	6	20	17	14	7
92	13	14	19	24	22	8	11
93	15	12	13	24	9	11	8
94	15	7	12	4	10	7	8
95	19	23	11	12	21	27	9
96	12	4	17	15	24	12	11
97	20	21	11	11	10	15	2
98	10	7	22	16	7	11	6
99	15	8	7	17	10	16	10
100	7	11	10	11	3	19	5
average	14.68	12.96	13.4	12.44	12.7	11.44	10.35

Fitness values obtain when choosing different percents for the Frozen Interval on reschedule interval = 8

run \ frozen	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1	4054.238	4539.58	4293.174	3704.882	4519.773	4323.135	3835.24	5526.058	4200.151	3551.49
2	3940.727	3408.566	3729.689	3666.992	3735.186	5229.36	4451.305	4334.918	4035.485	3977.292
3	3034.748	4130.133	4093.694	3881.193	4170.22	3983.723	4380.147	3714.933	5538.282	3858.48
4	3913.432	4308.009	4362.663	3831.75	3580.449	4209.019	4130.46	3896.72	4547.85	3903.673
5	4657.027	3586.995	4237.904	3970.51	3835.963	3489.443	3764.798	4039.118	4813.917	4400.175
6	3352.356	4941.901	4229.66	4263.74	3761.778	3837.305	4058.028	3365.222	3892.932	3874.051
7	3754.563	4238.621	4417.827	3856.964	4057.046	4229.835	4989.59	3723.245	3817.599	4430.51
8	4484.148	3761.011	4012.975	4189.301	3833.74	3624.514	4359.783	3843.667	4680.255	4442.144
9	4156.16	3913.504	3750.269	4136.876	3663.946	3916.253	3474.163	4668.642	3649.212	4002.469
10	4162.27	4115.629	4290.771	4066.357	3886.261	3805.775	3871.442	4354.757	2914.045	4892.426
11	2969.912	3752.294	3616.8	4523.451	4299.189	3852.249	4066.849	3846.152	4668.14	4503.467
12	4690.708	4529.185	4489.19	4053.88	3233.794	4483.154	4514.729	4801.018	3985.258	3815.846
13	4027.416	4442.606	3722.124	3621.957	3038.181	3927.514	4463.156	4036.318	3656.519	4493.14
14	3766.897	4106.466	4217.984	4163.586	3914.692	4126.293	3851.102	4916.291	3577.438	4237.514
15	3773.455	4296.706	3953.68	4769.35	3828.433	4223.253	3882.575	3971.981	3995.127	3826.482
16	4286.692	4328.839	4478.01	3969.856	3805.006	4386.639	4212.537	3966.029	5770.157	4290.146
17	3768.232	5003.135	3896.75	4244.858	3535.595	4084.443	3216.267	3981.985	3562.813	4389.412
18	3362.468	4206.374	3663.836	5153.832	3764.716	3946.096	4205.131	4737.715	4607	4361.77
19	4234.985	4346.779	4579.596	3743.564	3785.345	4102.147	3879.676	2893.513	4480.245	4660.99
20	4148.274	2687.173	4322.503	3600.881	3618.681	3689.324	4144.838	3811.135	4600.852	5299.293
21	4413.656	4311.652	3285.922	3849.719	3966.417	4158.155	3985.764	5015.905	3580.753	3925.491
22	4224.96	5241.078	4664.405	4395.927	3713.702	4112.108	4472.83	3728.22	4222.74	4821.818
23	4103.443	4142.977	4536.546	3785.67	3368.898	3976.5	4114.206	4054.539	4530.813	4747.792
24	4207.883	3889.201	3675.117	4075.935	3680.265	3774.29	3978.918	4467.007	4717.438	4027.448
25	3886.662	3773.481	4402.371	3900.58	4370.827	4534.098	4376.174	4820.769	3586.068	4419.256
26	4261.303	4209.318	3755.238	4388.088	2994.562	4323.038	4092.035	4425.539	3130.121	4230

run \ frozen	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
27	3660.058	3520.9	4082.965	3698.071	3514.643	4022.282	4239.396	3387.936	4427.995	4246.673
28	3932.876	3590.66	4109.281	4343.789	3083.226	4337.709	4547.899	4433.179	3385.329	4444.041
29	3170.272	3480.343	3954.084	4806.607	4116.765	4670.325	4366.036	3886.091	4127.328	4203.251
30	4095.205	3753.857	4554.133	4141.249	3202.307	4621.028	4008.588	3702.333	4292.017	3303.876
31	3776.186	3927.063	3636	3321.624	4829.868	4223.229	4213.007	4605.607	4297.268	4631.52
32	4686.148	3744.828	4238.838	3695.533	3616.068	4389.022	5432.801	4734.26	4714.515	3466.546
33	4309.313	3697.337	4375.925	3869.343	4610.328	4119.304	3984.553	3746.673	4980.524	4200.939
34	3633.634	3911.285	3769.059	3994.066	6043.297	3795.32	4423.47	4441.981	4081.868	4753.229
35	4297.083	3860.658	3882.676	3783.108	3894.846	4959.441	3488.832	4142.493	4612.871	4106.349
36	3419.769	5626	4246.209	3870.245	4095.143	4372.662	4766.212	4744.335	5220.078	4089.489
37	4044.174	3989.023	3665.832	3867.658	5366.578	3956.271	4342.601	3625.713	4052.779	4294.577
38	3563.669	4479.749	3822.258	4561.05	4184.493	4134.942	4354.442	3836.093	3904.682	4283.377
39	4099.434	3828.111	3925.758	4437.93	4323.555	4207.629	4486.458	3921.673	4666.407	3791.276
40	4218.689	4429.014	4716.892	4763.1	3971.182	3769.81	4372.764	4818.485	4910.85	3592.691
41	2882.819	3375.646	3656.906	3823.353	3678.552	3996.289	4289.486	4255.029	3409.489	3968.263
42	4590.024	4157.753	3512.42	3286.001	4967.203	4163.549	4071.248	3681.714	4100.845	3981.126
43	3707.736	4506.971	3615.578	4642.387	4072.016	4100.177	4767.42	4460.525	4149.51	5104.573
44	4143.074	3953.356	4612.742	4224.375	3789.486	3982.255	4058.394	4349.313	3880.018	4697.467
45	4129.286	4036.096	4014.76	4013.826	3877.642	4080.996	3752.953	3960.819	4922.485	4069.262
46	3612.164	3725.09	3840.203	3438.953	3111.552	4240.456	4168.927	3939.257	3510.389	3491.63
47	3836.955	2870.166	4265.112	3950.376	4058.265	4098.711	4717.26	4352.238	2916.355	4476.271
48	4084.021	4011.667	4597.114	4648.681	4497.841	3552.651	4329.18	3433.258	4644.224	4749.99
49	3116.507	4477.926	4169.043	4422.135	3725.458	4811.025	3644.743	3977.126	3581.98	4031.794
50	3608.206	4178.919	4359.797	4052.48	4154.241	4201.495	4286.505	4084.814	3605.426	4424.764
51	3190.729	3621.35	3666.356	4322.686	3833.424	4612.94	4093.645	4812.894	4259.324	3858.594
52	3673.16	4107.132	3779.777	3900.729	4758.49	4658.934	3590.913	3622.92	3991.992	4291.132
53	4598.444	3364.973	4422.411	4262.68	3775.764	4656.38	3980.313	4540.196	3951.793	4848.789
54	3076.084	3888.647	3751.96	3990.933	3871.225	4317.971	4422.61	4849.247	3839.051	4950.094
55	3477.358	3257.221	4639.901	3974.146	4071.915	3937.343	3972.366	4093.549	4543.03	3210.927

run \ frozen	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
56	4371.003	4122.78	5339.28	4315.729	3756.413	4077.466	4569.067	3607.299	4021.395	3825.623
57	4453.393	3217.233	4213.765	4390.039	3433.81	4372.764	4978.125	4582.563	4281.797	4890.854
58	3475.398	3896.059	4239.251	4803.493	4124.151	4196.477	3669.924	3748.617	4601.02	3743.27
59	3524.713	4190.33	4162.208	4195.456	4366.433	4291.826	3970.701	3841.369	4024.581	3858.802
60	3867.903	3790.987	4786.063	4124.501	4193.198	4709.123	4120.01	4911.871	3863.196	4764.06
61	4609.465	4266.755	4429.163	4291.589	3725.903	4433.155	3654.559	4613.76	4337.069	3869.347
62	4302.508	4005.918	4009.855	3613.234	4177.311	3736.131	4410.49	4373.776	4577.354	4483.786
63	3878.8	3666.527	4918.772	3959.477	4330.006	4090.68	3796.945	3721.37	4620.839	3555.374
64	4590.97	4197.988	4462.427	4608.025	3857.814	4038.471	4190.589	4619.255	3826.021	4535.98
65	4430.497	4020.053	3581.141	3873.178	4090.146	3479.75	3694.288	3994.146	5222.391	3640.01
66	4748.956	4466.503	3722.856	3702.762	3857.052	4087.923	3714.886	4221.112	4004.243	4019.939
67	3389.38	4196.237	3754.702	4541.311	4794.918	3786.72	4433.181	4221.506	4382.799	4175.18
68	4366.465	4012.053	3714.442	3950.467	4531.646	5153.242	3787.377	4060.167	5203	3650.04
69	3325.415	4063.567	4222.176	4283.963	4172.07	4351.067	3821.179	4433.427	4234.403	3602.885
70	4932.516	3529.243	3626.154	4020.268	4153.388	3920.061	3395.02	4565.491	4642.438	4818.17
71	3412.087	3302.46	4222.314	4059.898	4292.057	4145.748	4241.382	3990.198	3653.255	4876.198
72	4289.654	3826.814	4099.403	3588.006	3847.248	3986.49	3736.01	4128.575	3667.141	4277.29
73	4200.982	3863.408	5006.622	4443.139	4735.771	3293.67	3439.247	4120.827	4003.591	3881.09
74	3771.88	3718.772	3814.575	4350.793	3549.786	4369.621	3953.26	3281.515	5062.449	4047.204
75	3442.886	4475.675	3917.099	4677.815	3246.73	2999.561	4857.168	3984.499	4643.102	4807.676
76	4003.837	3457.636	3759.271	4617.583	4385.195	4022.625	3965.77	5055.557	4515.412	3757.742
77	4180.842	4354.141	4333.107	4258.337	4127.624	3639.466	3905.233	4830.867	4157.918	4131.258
78	5316.984	3478.304	3994.075	3519.781	4799.023	3454.151	4397.153	3672.771	4328.866	4546.396
79	3410.577	4220.9	3261.139	2946.5	3849	4611.901	4382.08	4078.913	4187.311	3684.229
80	4897.541	4004.898	4362.469	4470.684	3604.357	4292.356	3743.9	3250.563	4907.328	3957.5
81	3216.255	3964.947	4582.917	3661.99	4281.865	4230.485	4156.614	4001.272	3949.5	4670.825
82	3653.816	5330.296	4071.249	3319.036	3601.054	4739.02	4416.355	3995.401	4471.859	4461.029
83	4424.215	4991.615	4634.177	5184.242	3935.761	5205.76	3495.696	4285.344	3180.08	4400.115
84	3466.183	3453.47	3295.741	5307.099	4583.075	3688.371	4238.096	3711.28	3789.076	4790.03

run / frozen	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
85	3778.291	4148.003	4066.694	4785.249	3413.811	3399.933	3784.406	3904.181	3551.124	4370.505
86	5048.799	4998.711	3771.229	4152.673	3765.633	4321.842	4730.026	4117.358	3642.62	4422.871
87	3734.358	4152.621	3863.33	4230.876	4391.288	4048.32	4596.375	3641.471	3390.053	4045.192
88	3399.084	3966.705	4334.125	4130.068	4012.549	3600.524	4093.04	4436.432	4141.995	4106.206
89	3341.875	4046.472	3895.2	4115.25	4384.831	3825.579	5140.289	4119.67	5301.122	4973.884
90	3236.922	3727.885	3837.913	4105.653	4459.299	4299.83	3667.523	4765.807	4402.048	3697.959
91	3107.514	4428.965	4950.981	3674.314	3957.754	3901.176	4893.637	4146.447	4147.478	3841.757
92	3309.954	4325	4112.449	4240.796	4234.157	4303.229	4234.266	4453.777	4100.401	3753.297
93	4626.075	4214.375	4222.76	3831.896	3881.703	5067.323	4673.809	4076.885	4549.214	4313.625
94	3413.161	4187.34	3997.524	4627.156	5152.977	4158.409	4757.346	3367.881	2862.876	4900.077
95	3797.813	3745.768	3996.153	3962.394	4453.695	4359.563	4219.02	5068.99	3846.015	4769.721
96	3159.296	3838.865	4307.124	4551.25	4338.924	4013.603	3963.986	3702.673	5337.051	3522.4
97	3789.899	3624.95	4405.382	4414.466	4124.171	4487.268	3438.927	4407.305	3759.708	4716.895
98	3897.008	3686.628	4142.663	3967.288	4274.684	5035.594	4226.523	4095.464	4149.729	4139.038
99	3682.067	4353.781	3214.75	3761.176	4150.62	3518.385	4399.474	4030.828	4350.03	3502.017
100	3377.313	4828.613	4292.26	3833.311	3734.747	4387.273	3741.306	3888.869	4705.75	3794.508
average	3909.242	4039.432	4105.096	4113.79	4021.917	4154.697	4156.39	4155.085	4203.383	4215.369

بناء نظام تخطيط و جدولة متكامل و مرن و فعال لحل مشكلة التخطيط و الجدولة في الأنظمة متعددة العملاء باستخدام الخوارزميات الجينية

إعداد
هتـاف صـلاح الدين نـاطوريــة
المشرف
الأستاذ الدكتور نديم عبيد

المشرف المشارك
الدكتـور عمـاد صلاح

ملــخص

Arabic Summary

لقد تزايد في السنوات الأخيرة الاهتمام بتطوير أنظمة التخطيط و الجدولة المتعلقة بأنظمة الانتاج لما لها من تأثير كبير على جودة العمل و كفاءته, كما تزايد الاهتمام بنظرية تعدد العملاءً و اتسعت دائرة استخدامها في حل الكثير من المشاكل و خاصة المعقدة منها.

لقد تم في هذه الرسالة بناء نظام تخطيط و جدولة متكامل و مرن لحل مشكلة التخطيط والجدولة المتعلقة بأنظمة الإنتاج باستخدام الخوارزميات التطويرية و بالتحديد خوارزمية الجينات.

لقد تم اقتراح ثلاثة طرق بهدف زيادة فعالية و كفاءة نظام الجدولة المقترح في البحث الذي نشر عام 2007 من قبل الباحث "تشين". و من ثم تم حل المشكلة باستخدام عدة عملاء.

أثبت البحث كفاءته, حيث تحسنت النتائج بنسبة 0.25 في بعض الطرق المقترحة و 0.75 في بعضها الآخر و لم يثبت النظام متعدد العملاء كفاءته في جميع الحالات.

نقوم حالياً بتطبيق طرق تهدف لاستخدام النظام متعدد العملاء بشكل ذو كفاءة و فعالية أكبر في تحسين النتائج و التي تقوم على مقاييس : معيار كفاءة الجدولة أو التخطيط الذي تم التوصل إليه والوقت المستهلك و طول الكروموسوم و مدى استقرار بيئة الانتاج.