ESCUELA SUPERIOR POLIT ´

ECNICA DEL LITORAL



´

INSTITUTO DE CIENCIAS MATEM ATICAS

ESCUELA DE GRADUADOS

´

PROYECTO DE GRADUACI ON

´

PREVIO A LA OBTENCI ON DEL TITULO DE: MAG ´ISTER EN CONTROL DE OPERACIONES Y ´

GESTION LOG ´ISTICA

TEMA

USO DE METAHEURISTICAS PARA LA OPTIMIZACION DE  
LA SECUENCIA DE PRODUCCION Y LA ASIGNACION DE  
MANO DE OBRA EN UNA EMPRESA MANUFACTURERA

AUTORES:

H´ector Vinicio Sevilla Quevedo  
Carlos Francisco Zurita Noriega

Guayaquil -ECUADOR

A˜

NO 2010

**DEDICATORIA**

A mi hijo Mateo y a mi hija Romina, que en su tierna edad tambi´en contribuyeron al ´exito de la culminaci´on de este trabajo. Su inagotable amor me infundieron de aliento, paciencia y sobretodo de energ´ıa para llegar a la meta.

A mis padres por haberme dado la vida e inculcado el respeto y la responsabilidad, mi triunfo es su triunfo.

Finalmente quiero agradecer a mi esposa Mar´ıa Elena, por su inagotable paciencia durante el tiempo que duro el programa de Maestr´ıa y gracias a su permanente apoyo hoy puedo con orgullo ﬁnalizar esta etapa tan exitante de mi vida acad´emica y profesional. Solo puedo decirte que te amo.

*Vinicio Sevilla*

**DEDICATORIA**

A Myriam, la mejor mam´a del mundo, cuyo ejemplo, preocupaci´on, esfuerzo y apoyo m´as all´a de lo imaginable hicieron posible que yo pueda obtener este logro. Todo lo que obtengo es muy poco por m´ı y mucho debido a ti. No hay palabras mi viejita para agradecerte todo lo que has hecho y sigues haciendo por m´ı a pesar de que no lo merezca . . . Te amo mucho.

*Carlos Zurita*

**AGRADECIMIENTOS**

A mi familia por haberme apoyado siempre. A mi t´Nata(+) por su inﬁnito

ıa ˜amor, alegr´ıa y paciencia. A mis amigos de la Maestr´ıa Carlos, Santiago y Jorge, gracias por los buenos momentos compartidos.

A todas aquellas personas que me ayudaron a concluir este trabajo. Un agradecimiento especial a mi director de tesis, Fernando Sandoya, quien fue un guia y soporte en la elaboraci´on de este trabajo.

A Dios por haberme dado la salud para poder emprender y concluir todos los retos propuestos.

*Vinicio Sevilla*

**AGRADECIMIENTOS**

A mi abuela Elba, mi abuelo Manuel (+), mi tia Fabi y mi tio Carlos Augusto, gracias por haber sido un apoyo para mam´a en mi crecimiento y educaci´on.

A Jos´e Luis y Carlita, gracias por darme muchos momentos de alegr´ıa y cari˜no.

A Johana gracias por tu calidez y ternura mi amor.

A mis amigos de la Maestria, Vinicio y Santiago gracias por su ense˜nanza, apoyo y buenos momentos compartidos.

A mis mejores amigas de la vida, Alejandra y M´onica E. las mejores personas que alguien pueda conocer, gracias por su amistad.

A todos los amigos que a pesar de que soy ingrato siempre los recuerdo con mucho cari˜no, gracias por todo.

A mi Dios, gracias por haberme permitido conocer a tantas personas maravillosas con quien estar agradecido . . .

*Carlos Zurita*

**DECLARACION EXPRESA**

La responsabilidad por los hechos y doctrinas expuestas en este Proyecto de Graduaci´on, as´ı como el Patrimonio Intelectual del mismo, corresponde exclusivamente al ICM (Instituto de Ciencias Matem´aticas) de la Escuela Superior Politecnica del Litoral.

Ing. Vinicio Sevilla Quevedo Ing. Carlos Zurita Noriega

**TRIBUNAL DE GRADO**

MSc. Fernando Sandoya MSc. Washington Armas

DIRECTOR DE TESIS PRESIDENTE TRIBUNAL

M.A.E. Pedro Echeverr´ıa

VOCAL

**AUTORES DEL PROYECTO**

Ing. Vinicio Sevilla Quevedo Ing. Carlos Zurita Noriega

**INDICE GENERAL**

**DEDICATORIA I**

**AGRADECIMIENTO**S III

**DECLARACION EXPRESA V**

**TRIBUNAL DE GRADO VI**

**AUTORES DEL PROYECTO VII**

**INDICE DE FIGURAS XI**

**INDICE DE TABLAS XII**

**RESUMEN 1**

**OBJETIVOS GENERALES 3**

**OBJETIVOS GENERALES 4**

**INTRODUCCION 5**

**1 REVISION BIBLIOGRAFICA 7**

1.1 Planiﬁcaci´ondelaProducci´on..................... 7

1.2 Nacimiento de la Planiﬁcaci´on de la Producci´on . . . . . . . . . . . 8

1.3 ElNacimientodelosSistemasFormales. . . . . . . . . . . . . . . . 9

1.3.1 Laoﬁcinadecontroldeproducci´on . . . . . . . . . . . . . . 9

1.3.2 HenryGanttysudiagrama .................. 10

1.3.3 La planiﬁcaci´on basada en computadora . . . . . . . . . . . 11

1.3.4 Evoluci´on a los sistemas de Planiﬁcaci´on ﬁnita de la producci´on 12

1.4 Clasiﬁcaci´on de secuenciamiento de la producci´on . . . . . . . . . . 14

1.4.1 El Problema de secuenciamiento del Flow Shop (FSSP) . . . 15

1.4.2 El problema del secuenciamiento del Job Shop (JSSP) . . . 16

1.4.3 El problema del secuenciamiento del Open Shop (OSSP) . . 20

1.5 Complejidad computacional de los problemas de secuenciamiento de laproducci´on .............................. 20

1.6 Metodolog´ıas de soluci´on para los problemas de secuenciamiento de laproducci´on .............................. 22

1.6.1 AlgoritmosExactos ....................... 22

1.6.2 AlgoritmosdeAproximaci´on.................. 22

1.6.3 Metaher´ısticas.......................... 24

1.6.4 M´etodosH´ıbridos........................ 27

1.6.5 Metaheur´ısticas para el Job Shop Scheduling Problem . . . . 28

1.7 AlgoritmosGen´eticos.......................... 28

1.7.1 Espacio de B´usqueda en Algoritmos Gen´eticos . . . . . . . . 28

1.7.2 Conceptos B´asicos de los algoritmos gen´eticos . . . . . . . . 29

1.7.3 Estructura b´asica de un algoritmo gen´etico . . . . . . . . . . 30

1.7.4 Par´ametros de los algoritmos gen´eticos . . . . . . . . . . . . 30

1.7.5 Procesosdeunalgoritmogen´etico............... 32

1.8 Programaci´onenC++ ......................... 42

**2 DESCRIPCION Y FORMULACION DEL PROBLEMA 44**

2.1 M´etododePlaniﬁcaci´onActual .................... 44

2.2 Situaci´onrealdelaEmpresaABC. . . . . . . . . . . . . . . . . . . 45

2.2.1 Tiempo de Para en Celdas de Producci´on . . . . . . . . . . . 45

2.2.2 Horas extras pagadas no planiﬁcadas . . . . . . . . . . . . . 47

2.3 Notaci´onMatem´aticadelProblema . . . . . . . . . . . . . . . . . . 48

2.4 Estructura del algoritmo de soluci´on propuesto . . . . . . . . . . . . 49

2.4.1 Codiﬁcaci´ondelosGenes ................... 49

2.4.2 N´umerodeGeneraciones .................... 51

2.4.3 Tama˜nodelaPoblaci´on .................... 51

2.4.4 OperadordeMutaci´on ..................... 51

2.4.5 ProbabilidaddeMutaci´on ................... 52

2.4.6 OperadordeCruce ....................... 55

2.4.7 ProbabilidaddeCruce ..................... 55

2.4.8 ProbabilidaddeReemplazo .................. 56

2.4.9 ArchivodePar´ametros..................... 57

2.4.10 ArchivodeResultados ..................... 58

2.4.11 DiagramadeFlujo ....................... 58

**3 PRUEBAS Y RESULTADOS CON PROBLEMAS ESTANDAR 59**

3.1 ProblemasEst´andaryHardwareusado . . . . . . . . . . . . . . . . 59

3.2 Ejemplo de decodiﬁcaci´on y funcionamiento del algoritmo . . . . . . 60

**4 APLICACION EN EL PROBLEMA DE ESTUDIO 64**

4.1 Descripci´ondelProblemadeEstudio . . . . . . . . . . . . . . . . . 64

4.2 Resoluci´ondelProblemadeEstudio. . . . . . . . . . . . . . . . . . 67

4.3 Comparaci´on de la soluci´on propuesta con el m´etodo actual . . . . . 69

4.4 Asignaci´on de la mano de obra al secuenciameinto . . . . . . . . . . 70

4.5 Otrasoportundadesdemejora ..................... 70

4.5.1 Asignaci´on de mano de obra para balanceo de l´ıneas de producci´on ............................. 70

4.5.2 Cambioenelm´etododetrabajo. . . . . . . . . . . . . . . . 71

4.6 Implementaci´onenlaEmpresaABC . . . . . . . . . . . . . . . . . 72

4.6.1 Tiempo de Para en las celdas de producci´on . . . . . . . . . 72

4.6.2 Horas extra pagadas no planiﬁcadas . . . . . . . . . . . . . . 73

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4.7 Conclusiones y Recomendaciones | | | | | . . . . . . . . . . . . . . . . . . . | 73 |
| 4.7.1 | | Conclusiones | . . . . . . . . . . . . . . . . . . . . . . . . . . | | | 73 |
|  | 4.7.2 | Recomendaciones | | . . . . . . . . . . . . . . . . . . . . . . . . | | 75 |
| **ANEXO** |  |  | |  | | 77 |
| **BIBLIOGRAFIA** | | | | | | 82 |

**INDICE DE FIGURAS**

1.1 Un grafo disyuntivo representando un problema de Job Shop . . . . 18

1.2 Grafo disyuntivo para el c´alculo del makespam . . . . . . . . . . . . 19

1.3 Grafo disyuntivo para el c´alculo del makespam . . . . . . . . . . . . 20

1.4 Ejecuci´on general de un Algoritmo Gen´etico . . . . . . . . . . . . . 31

1.5 Representaci´on de la selecci´on por ruleta . . . . . . . . . . . . . . . 34

1.6 Situaci´onantesdelranking(ruleta) . . . . . . . . . . . . . . . . . . 35

1.7 Situaci´ondepu´esdelranking(ruleta) . . . . . . . . . . . . . . . . . 35

1.8 Cruceenunpunto ........................... 37

1.9 Cruceendospuntos .......................... 37

1.10Cruceendospuntos .......................... 37

1.11CruceUniforme ............................. 38

1.12CruceAritm´etico ............................ 38

1.13PMX:Lospadresson .......................... 39

1.14PMX:Loshijosson ........................... 39

1.15PMX:Completarloshijos ....................... 39

1.16 PMX:Semapeanlosvaloresrestantes . . . . . . . . . . . . . . . . 40

1.17PMX:Ejemploadicional ........................ 40

1.18JOX ................................... 41

1.19Inversi´ondeGenes ........................... 41

1.20CambiodeOrden ............................ 41

2.1 Diagrama de Pareto para los problemas de Productividad . . . . . . 46

2.2 Grafo disjunto que representa el problema del Job Shop . . . . . . . 50

2.3 Mutaci´onSimple ............................ 52

2.4 Desempe˜no de la calibraci´on del porcentaje de mutaci´on . . . . . . 54

2.5 Desempe˜no de la calibraci´on del porcentaje de cruce . . . . . . . . . 56

2.6 DiagramadeFlujo ........................... 58

3.1 Codiﬁcaci´ondelproblema ....................... 60

3.2 Soluci´ondelproblema ......................... 61

3.3 Makespam ................................ 63

3.4 Secuenciamiento............................. 63

4.1 DiagramadeGanttdelasecuencia .................. 68

**INDICE DE TABLAS**

1.1 Principales diagramas creados por Gantt . . . . . . . . . . . . . . . 10

1.2 Datosparac´alculodelmakespan ................... 19

1.3 Reglasdedespachom´ascomunes ................... 23

1.4 Representaci´onBinaria ......................... 33

1.5 Representaci´on basada en Permutaciones . . . . . . . . . . . . . . . 33

1.6 Representaci´onDirecta ......................... 33

1.7 Crucedetrespadres .......................... 38

2.1 Tiempodeparapromediodiario .................... 45

2.2 Tiempodeparapromediodiario .................... 47

2.3 Horasextranoplaniﬁcadaspormes .................. 48

2.4 Unasoluci´onalproblemaplanteado .................. 50

2.5 Tabla de calibraci´on de probabilidad de mutaci´on . . . . . . . . . . 53

2.6 Tabla de calibraci´on de probabilidad de cruce . . . . . . . . . . . . 55

2.7 Tabla de calibraci´on de probabilidad de reemplazo . . . . . . . . . . 57

3.1 Mejoresresultadosporinstancia .................... 60

4.1 Problema a resolver con el algoritmo propuesto . . . . . . . . . . . 65

4.2 Codiﬁcaci´ondelasTareas ....................... 67

4.3 Resultadosdelasoluci´onalproblema ................. 67

4.4 Secuenciaci´ondelasTareas ...................... 69

4.5 Resultados de la soluci´on mediante el m´etodo actual . . . . . . . . . 69

4.6 TiempodeParapromediodiario.................... 72

4.7 Horasextranoplaniﬁcadaspormes .................. 73

**RESUMEN**

La presente investigaci´on ha sido desarrollada para optimizar las actividades de secuenciamiento de producci´on y asignaci´on de mano de obra en una empresa espec´ıﬁca, que por cuestiones de conﬁdencialidad de la informaci´on usada denominaremos empresa ABC. Actualmente, en la Empresa ABC la Planiﬁcaci´on Finita de Producci´on, el Secuenciamiento de la Producci´on y la asignaci´on de mano de obra a las diferentes tareas se realiza manualmente. Esta tarea toma aproximadamente 2 d´ıas de trabajo a la semana de los siguientes elementos de la direcci´on de la empresa: el Jefe de Planiﬁcaci´on, el Jefe de Producci´on y el Gerente de Producci´on. Las principales restricciones que se toman para la planiﬁcaci´on son:

N´umero de personal disponible

Ingreso de materia prima fresca

Horas hombre necesarias por d´ıa

Debido a que se emplean m´ultiples recursos es importante que la Planiﬁcaci´on sea resuelta de manera ´optima, de forma que se puedan ahorrar todos los recursos asociados al problema. Para poder resolver el problema de manera ´optima en esta tesis, debido a que se trata de un problema NP-duro, se plantea un algoritmo gen´etico, que implementado en el lenguaje de prop´osito general C++, permite resolver el problema de secuenciaci´on de ´ordenes de trabajo, en ingl´es: Job Shop Scheduling (JSSP). El algoritmo propuesto se fundamenta en el principio de selecci´on natural y utiliza mutaci´on clonal, y se demostr´o que funciona de manera eﬁciente en la resoluci´on de los problemas de prueba planteados en esta tesis. Tambi´en se incorpora un mecanismo que permite explorar la vecindad con la selecci´on de la mejor soluci´on, de esta manera se logra una mejora signiﬁcativa de resultados.

***ICM 1 ESPOL***

Adicionalmente para probar la validez del algoritmo construido se prueba resolviendo problemas est´andar encontrados en la literatura del secuenciamiento de producci´on. De estas pruebas se concluye que el algoritmo propuesto tarda m´as tiempo en resolver los problemas est´andar que los m´etodos actuales, sin embargo para resolver el problema propuesto en la Empresa ABC toma aproximadamente 3 minutos obteni´endose una mejor soluci´on a la actual y permitiendo que las personas de planiﬁcaci´on y producci´on puedan dedicar su tiempo a otras tareas.

**OBJETIVOS GENERALES**

El objetivo principal de ´esta investigaci´on es desarrollar una metaheur´ıstica que resuelva el problema de secuenciamiento de la producci´on y asignaci´on de mano de obra en una empresa manufacturera. Para ello se ha investigado el estado del arte en los problemas de secuenciamiento de la producci´on (Job Shop Scheduling Problem). con el ﬁn de encontrar la metaheur´ıstica m´as apropiada para resolver el problema planteado.

El algoritmo desarrollado tiene que ser mucho m´as eﬁciente que el m´etodo tradicional usado que consiste en una planiﬁcaci´on manual basada en la experiencia de las personas del ´area de producci´on de la empresa.

**OBJETIVOS PARTICULARES**

Los objetivos particulares de la presente investigaci´on son:

Desarrollar un algoritmo gen´etico que secuencie la producci´on de la empresa ABC.

Calibrar los par´ametros del algoritmo gen´etico usando una t´ecnica experimental de aproximaci´on de un solo factor.

Disminuir las horas extras no planiﬁcadas e improductivas con un mejor secuenciamiento de la producci´on.

Facilitar la planiﬁcaci´on para el responsable en la Empresa ABC de forma que sea m´as f´acil controlar la programaci´on de piso.

Poder manejar escenarios y hacer cambios de planes en funci´on de la informaci´on real del cumplimiento de planiﬁcaci´on.

**INTRODUCCION**

La mejora de los procesos productivos en las empresas manufactureras ha sido un factor clave para el ´exito de los negocios. Estos procesos se han vuelto muy complicados debido a que son procesos sist´emicos en los que el objetivo a cumplir es fabricar una cierta cantidad determinada de un mix de productos deseados al menor costo posible con recursos limitados y en un rango de fechas requeridas. En estos procesos intervienen recursos como personal, m´aquinas, materia prima e insumos.

El objetivo es determinar el Plan de Producci´on y ejecutarlo en una secuencia tal que se minimicen tiempos improductivos, tardanza en las ´ordenes, u otros criterios deseados. Adicionalmente hay que tomar en cuenta que los recursos que se utilizan en las diversas operaciones productivas son limitados y costosos. El problema se complica dado que existen com´unmente dependencias entre las diversas operaciones. De ´esta forma resulta relevante para los sistemas de manufactura, el estudio de los problemas de planiﬁcaci´on o scheduling en los cuales se trata de determinar la secuencia y las fechas exactas en las cuales deben ser realizadas las tareas planiﬁcadas.

Dentro del ´area del Scheduling, el problema del Job Shop (JSSP) es el que m´as aplicaci´on pr´actica ofrece, pues permite incrementar la eﬁciencia de los procesos de manufactura. Desde los primeros a˜nos de la d´ecada de 1960, los problemas de Job Shop, han sido de mucho inter´es para los investigadores en m´etodos de optimizaci´on, ya que son problemas dif´ıciles del tipo NP-duro. En el pasado se han desarrollado un sinn´umero de Heur´ısticas espec´ıﬁcas para este problema. De ´estas, el procedimiento de sustituci´on del cuello de botella (Shifting Bottleneck Procedure) se convirti´o en uno de los m´as usados. Hace pocos a˜nos el desarrollo de nuevas t´ecnicas universales

***ICM 5 ESPOL***

de b´usqueda han tenido auge gracias a los avances computacionales y podr´ıan convertirse en m´etodos satisfactorios para resolver el problema del job shop (JSSP). Los Algoritmos Gen´eticos, el Recocido Simulado, B´usqueda TABU, B´usqueda Variable Local Paralela entre otras son las nuevas t´ecnicas que se est´an utilizando para poder resolver el problema del Job Shop.

El objetivo principal de ´este estudio es encontrar un algoritmo que permita encontrar una mejor soluci´on al m´etodo actualmente propuesto y sin ocupar el tiempo valioso del personal de Producci´on y Planiﬁcaci´on de la Empresa ABC.

**CAPITULO 1**

**REVISION BIBLIOGRAFICA**

**1.1. Planiﬁcaci´on**

**on de la Producci ´**

Oliver Wight quien es considerado actualmente un experto en planiﬁcaci´on de la producci´on, debido al concepto propuesto de excelencia operacional a trav´es del plan de ventas y operaciones, estableci´o que la planiﬁcaci´on de producci´on debe tomar en cuenta dos problemas principales los cuales son *prioridade*s y *capacidad*. En otras palabras determinar la secuencia de producci´on y en que m´aquina se har´a.

Cabe indicar que en las empresas de producci´on existen m´ultiples tipos de planiﬁcaci´on, incluyendo la planiﬁcaci´on de ventas y operaciones, el plan maestro de producci´on, la planiﬁcaci´on de capacidad, la planiﬁcaci´on de requerimiento de materiales, la planiﬁcaci´on de recursos empresariales, la planiﬁcaci´on y control de manufactura y la planiﬁcaci´on ﬁnita o secuenciamiento detallado en el piso de producci´on. Este ´ultimo tipo de planiﬁcaci´on indica espec´ıﬁcamente los tiempos de inicio y ﬁnalizaci´on de cada una de las operaciones en los centros de trabajo existentes, adem´as es el m´as complejo debido a todas las restricciones que deben ser analizadas para obtener un secuenciamiento ´optimo de producci´on.

“La Planiﬁcaci´on ﬁnita se deﬁne como la asignaci´on real de fechas de inicio y ﬁnalizaci´on de las operaciones o grupo de operaciones para indicar cuando deber´ıan ser ejecutadas para completar la orden de producci´on a tiempo”1

1Cox et al. APICS Dictionary. 1992 17 pag.

***ICM 1-7 ESPOL***

**1.2. Nacimiento de la Planiﬁcaci ´on**

**on de la Producci ´**

Con el nacimiento de la Revoluci´on Industrial en la segunda mitad del siglo XVIII, aparecieron formalmente talleres centralizados donde se fabricaban los productos requeridos. Esta Revoluci´on Industrial provoc´o cambios socioecon´omicos, tecnol´ogicos y culturales en la humanidad, debido principalmente a que la econom´ıa basada en el trabajo manual fue reemplazada por la basada en el trabajo industrial. Estos talleres fueron el inicio de las f´abricas modernas. Los primeros talleres eran peque˜nos, en los cuales se fabricaba pocos productos en grandes lotes. Las mejoras de productividad proven´ıan de disminuir las operaciones de calibraci´on que consum´ıan mucho tiempo por lo que no hab´ıa un enfoque en la planiﬁcaci´on de producci´on. Al ﬁnalizar el siglo XIX las empresas manufactureras se empezaron a preocupar de maximizar la productividad de los costosos equipos de la f´abrica. Es decir el principal objetivo era mantener la utilizaci´on de l´ınea (producci´on real versus producci´on te´orica en porcentaje) en niveles altos, para esto los administradores empezaron a dar importancia a la planiﬁcaci´on de producci´on.

“Al Inicio de la Epoca Industrial la Planiﬁcaci´on, cuando era usada, consist´ıa ´on del trabajo y no propor

unicamente de un listado del inicio y ﬁnalizaci´cionaba ninguna informaci´on de la duraci´on total de cada orden o del tiempo requerido para completar las operaciones individuales”2

El listado de inicio y ﬁnalizaci´on de trabajo fue ampliamente usado antes del nacimiento de los m´etodos formales de planiﬁcaci´on (aunque todav´ıa puede ser encontrado en algunas f´abricas); en otras palabras la planiﬁcaci´on de la producci´on era efectuada por una persona con t´ecnicas muy rudimentarias y b´asicas para el efecto. Cabe indicar que los m´etodos informales como el basado en el despacho todav´ıa no han desaparecido.

“La administraci´on de inventario y producci´on en muchas compa˜n´ıas actual-mente consiste ´ordenes y despachar”3

unicamente en lanzar ´

2Roscoe y Freark. Organization for Production. 1971. 15 pag.

3Wight, O. Production and Inventory Management in the Computer Age.1984.25 pag.

***ICM 1-8 ESPOL***

**1.3. El Nacimiento de los Sistemas Formales**

Para el principio del siglo XX todo cambi´o, debido a que las empresas manufactureras empezaron a fabricar una amplia variedad de productos, la misma que increment´o la complejidad a los administradores. La administraci´on cient´ıﬁca fue la respuesta para mejorar el control de esta complejidad. Las f´abricas crecieron signiﬁcativamente, con lo cual la reducci´on de costos fue por primera vez el objetivo fundamental de los responsables. Por estas razones empez´o el trabajo de alcanzar econom´ıas de escala y costos unitarios de producci´on bajos.

**1.3.1. La oﬁcina de control de producci ´on**

Inicialmente los operarios planeaban libremente las actividades de su trabajo seg´un la forma en que ellos cre´ıan conveniente. Frederick Taylor propuso que los administradores realicen la planiﬁcaci´on de actividades, de ´esta manera separ´o la planiﬁcaci´on de la ejecuci´on de la producci´on con lo que se justiﬁc´o el uso de m´etodos formales de planiﬁcaci´on y la creaci´on de la oﬁcina de control de producci´on.

La ﬁlosof´ıa de Taylor se fundamentaba en tres aspectos:

Las leyes cient´ıﬁcas gobiernan cuanto puede producir un trabajador por d´ıa.

Es funci´on de la gerencia descubrir y utilizar estas leyes en la operaci´on de sistemas productivos.

Es funci´on del trabajador cumplir los deseos de la gerencia sin cuestionarlos.

A partir de los postulados de Taylor, las gerencias requirieron personal para planiﬁcar, administrar el inventario y monitorear las operaciones. La oﬁcina de control de producci´on creaba un plan maestro de fabricaci´on basado en las ´ordenes en ﬁrme y en la capacidad de la planta. El objetivo principal de esta oﬁcina era crear ´ordenes de trabajo en funci´on de los requerimientos m´as urgentes y abastecer de material al piso de producci´on.

***ICM 1-9 ESPOL***

**1.3.2. Henry Gantt y su diagrama**

Debido a que a´un exist´ıa mucha informalidad y no se ten´ıan m´etodos estandarizados entre las oﬁcinas de producci´on de las diferentes f´abricas, Gantt cre´o un diagrama innovador, que lleva su nombre, para el control de la producci´on (1). El diagrama de Gantt es una herramienta visual para indicar la secuencia de los trabajos que ser´an efectuados en un conjunto de m´aquinas y para monitorear su avance.

Gantt cre´o muchos tipos de diagramas que representan diferentes formas de estudiar los sistemas de producci´on y analizar m´etricas. Adem´as dise˜n´o sus diagramas de forma que los planiﬁcadores pudiesen r´apidamente conocer el uso ideal y el uso real de los recursos a trav´es del tiempo.

Gantt cre´o muchos tipos de diagramas que representan diferentes formas de estudiar los sistemas de producci´on y analizar m´etricas. Adem´as dise˜n´o sus diagramas de forma que los planiﬁcadores pudiesen r´apidamente conocer el uso ideal y el uso real de los recursos a trav´es del tiempo.

Tabla 1.1: *Principale*s *diagrama*s *creado*s *po*r *Gant*t

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo de diagrama | Unidad | Cantidad que va a ser medida | Representaciones de tiempo | Fuentes |
| Balance diario de trabajo | Parte u operaci´on | Cantidad producida | Filas para cada d´ıa;las barras muestran datos de inicio y datos de f´ın. | Gantt, 1903; Rathe, 1961 |
| Registro de personal | Trabajador | Cantidad de trabajo hecho cada d´ıa y semana, medidos como tiempo | 3 o 5 columnas por cada d´ıa en dos semanas | Gantt, 1981; Rathe, 1961 |
| Registro de maquinaria | M´aquina | Cantidad de trabajo hecho cada d´ıa y semana, medidos como tiempo | 3 o 5 columnas por cada d´ıa en dos semanas | Gantt, 1919, 1981; Rathe, 1961 |
| Diagrama dee ubicaci´on | M´aquina | Avance de tareas asignadas, medidas como tiempo | 3 o 5 columnas por cada d´ıa en dos semanas | Clark, 1942 |
| Diagrama de trabajo de Gantt | Tipo de m´aquina | Tareas programadas y el total de trabajo por fecha | Una columna por cada d´ıa para dos meses | Mitchell, 1939 |
| Diagrama de trabajo de Gantt | Orden de trabajo | Trabajo completado para una fecha, medido como tiempo | Una columna por cada d´ıa para dos meses | Mitchell, 1939 |
| Diagrama de secuenciamiento | Tareas en un trabajo | Inicio y ﬁn de cada tarea | El eje horizontal dividido en 45 d´ıas | Muther, 1944 |
| Diagrama de avance | Producto | Cantidad producida cada mes | 5 columnas para cada mes para cada a˜no | Gantt, 1919, 1981; Rathe, 1961 |
| Diagrama de ´ordenes de trabajo | Orden de trabajo | Cantidad producida cada mes | 5 columnas para cada mes para cada a˜no | Gantt, 1919, 1981; Rathe, 1961 |

**1.3.3. La planiﬁcaci ´on basada en computadora**

A pesar que la planiﬁcaci´on de la producci´on era una tarea compleja, los esfuerzos iniciales de usar algoritmos computacionales se centraron en la planiﬁcaci´on de grandes proyectos, principalmente de construcci´on o desarrollo de sistemas. La planiﬁcaci´on de la producci´on basada en la computadora comenz´o en la segunda mitad de la d´ecada de los 60 aproximadamente.

Existen tres factores que contribuyeron al uso de computadoras para la planiﬁcaci´on de la producci´on, estos son:

IBM desarroll´o el sistema de control e informaci´on de la producci´on en 1965.

La implementaci´on de este sistema y otros similares guiaron al conocimiento pr´actico del uso de computadoras.

Los investigadores compararon estas experiencias y desarrollaron nuevas ideas en la administraci´on de la producci´on.

Los primeros sistemas computacionales de planiﬁcaci´on fueron proyectos de investigaci´on realizados en universidades los cuales usaron computadoras centrales y terminales. Las terminales recib´ıan datos acerca de la ﬁnalizaci´on de las tareas o el movimiento de material; la computadora central actualizaba la informaci´on, inclu´ıa registros para las m´aquinas y empleados, materiales consumidos y colas en las estaciones de trabajo. A partir de esta informaci´on, la computadora creaba para cada estaci´on una lista de despacho con las tareas pendientes que deb´ıan ser procesadas. Para crear una lista de despacho, el sistema usaba una regla que consideraba uno

o m´as factores incluyendo el tiempo de proceso, fecha de ﬁnalizaci´on, tardanza, n´umero de operaciones restantes o costos de producci´on.

La planiﬁcaci´on basada en computadora evolucion´o desde proyectos de investigaci´on hacia sistemas comerciales. Uno de los primeros sistemas comerciales fue el Sistema de Administraci´on Log´ıstica (LMS por sus siglas en ingl´es), desarrollado por IBM para sus instalaciones de producci´on de semiconductores. El sistema LMS comenz´o a operar alrededor de 1980 como una herramienta para modelar los recursos productivos. Los m´odulos capturaban informaci´on del piso, recuperaban las prioridades del

***ICM 1-11 ESPOL***

plan de producci´on diario y ﬁnalmente tomaba decisiones de despacho.

En conclusi´on el sistema ten´ıa dos objetivos b´asicos:

Soporte pasivo a las decisiones: Proporcionando informaci´on detallada a los usuarios acerca de la producci´on; y,

Despacho proactivo: Proporcionando sugerencias de producci´on en funci´on de reglas establecidas con anterioridad.

Este software us´o estrategias est´andar para generar una planiﬁcaci´on sugerida que luego pod´ıa ser modiﬁcada por el personal de acuerdo a sus necesidades. La ventaja m´as grande de usar computadoras es la reducci´on dr´astica de tiempo usado en generar un plan de producci´on. De esta forma se desarrollaron los sistemas comerciales que planiﬁcan y controlan la producci´on.

A medida que los sistemas basados en computadora se volv´ıan populares los programadores mejoraron su funcionalidad enfoc´andose ﬁnalmente en sistemas que pod´ıan tomar decisiones por s´ı solos.

**1.3.4. Evoluci ´on ﬁnita de la producci ´on a los sistemas de Planiﬁcaci ´on**

La aplicaci´on de las computadoras se extendi´o a otras funciones de planiﬁcaci´on de la producci´on, las cuales se listan a continuaci´on en orden cronol´ogico:

La Planiﬁcaci´on de Requerimiento de Materiales (MRP): Traslada la demanda de art´ıculos terminados en una planiﬁcaci´on basada en el tiempo de compra y de producci´on de insumos y semielaborados respectivamente.

La Planiﬁcaci´on de Requerimientos Empresariales (ERP): M´etodo para la planiﬁcaci´on y el control sincronizado de todos los recursos involucrados en la cadena de valor de una empresa. Los ERP se componen generalmente de m´odulos como Contabilidad, Finanzas, Manufactura, Producci´on, Transporte, Ventas y Distribuci´on, Recursos Humanos y Materiales.

La Planiﬁcaci´on Finita o Secuenciamiento de Producci´on: Toma en cuenta las condiciones reales del piso incluyendo restricciones de capacidad, herramientas,

***ICM 1-12 ESPOL***

personal, fechas de entrega comprometidas y los requerimientos de las ´ordenes que ya han sido lanzadas. Esta planiﬁcaci´on es una pieza clave del sistema ERP II, mismos que incluyen adem´as Gesti´on de Cadena de Suministro (SCM), Gesti´on de Relaciones con el Cliente (CRM) y e-business.

Los Sistemas Finitos tienen 3 componentes primarios que son:

Base de datos.

Motor de planiﬁcaci´on

Interfaz de usuario.

El sistema de planiﬁcaci´on puede compartir una base de datos con otro sistema (como un MRP) o tener su base propia la cual se actualiza autom´aticamente desde el otro sistema. La interfaz de usuario ofrece diferentes formas de ver la planiﬁcaci´on, entre las m´as importantes est´an los diagramas de Gantt, listas de despacho y gr´aﬁcos de uso de recursos. El motor de planiﬁcaci´on genera sugerencias para lo cual usa planiﬁcaci´on basada en reglas, optimizaci´on o simulaci´on.

Los sistemas de Planiﬁcaci´on Finita de Producci´on se han vuelto una herramienta fundamental para el ´exito de las empresas manufactureras. De hecho se ha establecido el uso obligatorio de estos sistemas para poder obtener una ventaja competitiva que permita a las empresas actuales sobrevivir a la dura competencia y a los cambios en los factores externos que afectan la rentabilidad.

A pesar de la importancia de los Sistemas de Planiﬁcaci´on Finita de Producci´on solamente el 30 % de las Empresas en Estados Unidos los usan como parte de sus operaciones. En pa´ıses del tercer mundo la cifra se reduce a incluso menos del 5 %, lo cual demuestra que en estos pa´ıses existe una gran oportunidad de mejora.

Los factores determinantes para que las empresas no usen ´este sistema de planiﬁcaci´on son:

Desconocimiento de la existencia y los beneﬁcios que ´este sistema brinda. Adicionalmente, falta de bases matem´aticas en optimizaci´on combinatoria para poder desarrollar un sistema dentro de la propia empresa.

***ICM 1-13 ESPOL***

Costos de implementaci´on altos y accesibles solo a grandes empresas. Una implementaci´on cuesta en promedio 150.000 USD teni´endose programas que pueden llegar incluso al mill´on de d´olares.

Tiempos de implementaci´on altos, toma en promedio 4 meses.

Pocos proveedores quienes ofrecen sistemas compatibles s´olo con los ERP de las mismas ﬁrmas desarrolladoras. Para usar un ERP de una ﬁrma y un secuenciador de otra se requerir´an desarrollar las interfases necesarias; esto se traduce en mayores costos y tiempos de implementaci´on.

La mayor desventaja es que estos sistemas son verdaderas *caja*s *negras*, es decir el usuario no sabe el algoritmo que est´a usando el sistema para encontrar la soluci´on. Debido al desconocimiento en optimizaci´on combinatoria, el usuario no puede demostrar la validez de la respuesta. En conclusi´on no se conoce si se est´a comprando a alto precio un sistema antiguo que no genera buenas soluciones.

**1.4. Clasiﬁcaci´on**

**on de secuenciamiento de la producci ´**

Con el inter´es de los investigadores enfocado en encontrar una Planiﬁcaci´on Finita de la Producci´on ´optima, se propusieron varios problemas est´andar de secuenciamiento de la producci´on. Debido a la gran cantidad de entornos productivos existentes se plantearon una variedad de problemas, muchos de los cuales se volvieron te´oricos sin aplicaci´on pr´actica pero fueron la base para desarrollar m´etodos de soluci´on a la planiﬁcaci´on ﬁnita.

Todos los problemas de secuenciamiento de producci´on tienen caracter´ısticas comunes, sin embargo se los puede clasiﬁcar bajo los siguientes criterios:

Proceso de llegada de las tareas: Divididos en est´aticos (se conoce exactamente el inicio de las tareas) o din´amicos (el inicio de las tareas depende de una funci´on probabil´ıstica).

Pol´ıticas de inventario: Divididos en abiertos (los productos siguen la pol´ıtica de producci´on bajo pedido) o cerrados (los productos siguen una pol´ıtica de

***ICM 1-14 ESPOL***

producci´on para inventario). En la vida real se encuentran sistemas h´ıbridos (mezcla de las dos pol´ıticas).

Atributos de las tareas: Pueden ser determin´ısticos (los tiempos de proceso y la disponibilidad de m´aquinas es conocida previamente) o estoc´asticos (no se cumple lo anterior).

Conﬁguraci´on de la producci´on: Esto depende si la producci´on es de un solo paso o de m´ultiples pasos. En el primer caso cada tarea se hace en una m´aquina mientras que en el segundo caso, cada tarea consta de una serie de operaciones que pueden ser procesadas en diferentes m´aquinas.

Otra clasiﬁcaci´on y con seguridad la m´as usada por los Investigadores Te´oricos es la basada en el ﬂujo de tareas a trav´es de los recursos, dependiendo de esto se clasiﬁcan en:

Problemas de secuenciamiento de ﬂow shop.

Problemas de secuenciamiento de job shop.

Problemas de secuenciamiento de open shop.

Otros problemas de secuenciamiento.

**1.4.1. El Problema de secuenciamiento del Flow Shop (FSSP)**

El Flow Shop Scheduling Problem es un problema de secuenciamiento de la producci´on que de forma general se describe as´ı:

Se tiene un conjunto de N tareas j =(*J*1*,J*2*,J*3*,...,Jn*). que tienen que ser procesadas en un conjunto M de m´aquinas m =(*M*1*,M*2*,M*3*,...,Mm*). En el FSSP general todas las tareas J deben ser procesadas secuencialmente en todas las m´aquinas *M*, en el mismo orden. Se asume que una operaci´on se inicia solamente si se ha terminado la anterior y que los tiempos de cambio y calibraci´on est´an incluidos en los tiempos de proceso. Adicionalmente, cada tarea debe ser procesada en sola-mente una m´aquina a la vez, y cada m´aquina puede procesar una tarea a la vez. El problema consiste en encontrar la secuencia de tareas factible que optimice un

***ICM 1-15 ESPOL***

criterio espec´ıﬁco determinado (tiempo total de proceso, n´umero de tareas atrasadas

o tiempo de tardanza total). Existen muchas variaciones del FSSP como:

El piso de producci´on con cero almacenamiento: Una tarea j que ha sido procesada en una m´aquina *M* i no puede avanzar a la m´aquina *Mi*+1 hasta que ´esta termine de procesar la tarea predecesora a *j*. En ´este caso la tarea j permanece en la m´aquina *Mi*, retrasando tambi´en el trabajo de ´esta m´aquina.

El piso de producci´on con bloqueo: Es similar a la anterior pero en ´este caso puede existir un almacenamiento intermedio.

Aunque todas estas variantes del FSSP tienen muchas aplicaciones industriales los investigadores se han enfocado principalmente en una versi´on reducida del problema FSSP general, donde se asume que las tareas deben ser procesadas en la misma secuencia por cada una de las M m´aquinas.

**1.4.2. El problema del secuenciamiento del Job Shop (JSSP)**

El Job Shop Scheduling Problem es un problema de secuenciamiento de la producci´on que de forma general se describe as´ı: Se tiene un conjunto de m´aquinas m =(*M*1*,M*2*,M*3*,...,Mm*) las cuales deben realizar un conjunto de tareas j = (*J*1*,J*2*,J*3*,...,Jn*). En el JSSP general, cada tarea j debe ser procesada en una m´aquina a la vez y requiere de una serie de operaciones *O*1*j,O*2*j,...,Onj*. Estas operaciones deben ser secuenciadas en un orden predeterminado diferente para cada tarea. Se tiene como restricci´on que una operaci´on se inicia solamente si se ha terminado la anterior (*Oi*j *−→ Oi,j*+1) y que los tiempos de cambio y calibraci´on est´an incluidos en los tiempos de proceso que son conocidos previamente *pij*. Adicionalmente, cada tarea debe ser procesada en solamente una m´aquina a la vez, y cada m´aquina puede procesar una tarea a la vez. La dimensi´on de una instancia de JSSP es n x m, donde n es el n´umero de tareas a realizar y m el n´umero de m´aquinas disponibles. El problema radica en encontrar una secuencia de operaciones factible para cada m´aquina de forma que se optimice un criterio espec´ıﬁco, generalmente el tiempo total de proceso. Inicialmente, las soluciones del JSSP se representaron

***ICM 1-16 ESPOL***

mediante Diagramas de Gantt, lamentablemente estos diagramas no indican las relaciones existentes entre actividades; estas restricciones llevaron al desarrollo de otros modelos de representaci´on. Se desarrollaron Diagramas de Gantt m´as avanzados como el Diagrama de Gantt en tres dimensiones (tareas, m´aquinas y tiempo) para mejorar la representaci´on de los problemas de secuenciamiento de la producci´on.

Otro m´etodo de representaci´on y el m´as usado actualmente es la representaci´on mediante grafos disyuntivos. Un grafo disyuntivo est´a dado por: G =(*V*, *A*, *E*), donde:

G: Grafo disyuntivo.

V: Conjunto de nodos del grafo que representa las operaciones a ser procesadas en el conjunto de m´aquinas *m*.

A: Conjunto de arcos conjuntivos que representan las restricciones de procedencia entre las operaciones de cada tarea, de forma que (*Okl,Okm*)*�A*, indica que la operaci´on O*k*l es la inmediata predecesora de la operaci´on *Okm*.

E: Conjunto de arcos dirigidos y que representan las restricciones de capacidad, enlazando un par de arcos disyuntos que comparten una misma m´aquina. As´ı, dos operaciones *Okl,Oi*l que son ejecutadas en la misma m´aquina l no pueden hacerlo simult´aneamente.

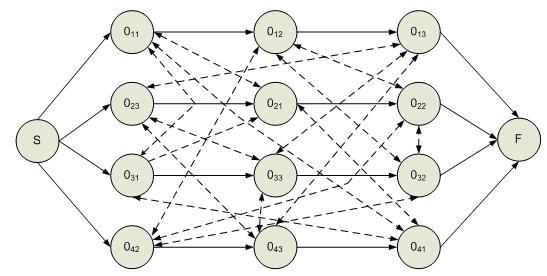
El nodo inicial ﬁcticio S es llamado la fuente y el nodo ﬁcticio F es llamado sumidero. El peso positivo para cada nodo representa el tiempo de procesamiento *dkl*. El nodo inicial se conecta con la primera operaci´on de cada tarea, mediante un arco que tiene por longitud el tiempo de liberaci´on (*ri) y de igual forma la ultima operaci´*

´on de cada tarea se conecta con el nodo ﬁnal. El objetivo del problema es minimizar el tiempo total de proceso satisfaciendo las restricciones mediante la eliminaci´on de ciertos arcos disyuntivos y convertir los que quedan en conjuntivos seg´un el m´etodo de secuenciamiento que se quiera utilizar. En la Figura 1.1 se presenta un grafo disyuntivo para un problema de job shop (4 × 3) (cuatro tareas y tres m´aquinas).

Con estos argumentos se puede deﬁnir el JSSP matem´aticamente de la siguiente manera:

Min. *tn*(camino mas largo entre S y F)

***ICM 1-17 ESPOL***



S.A.

*tk*m + *tk*l ≥ *dk*l

*tk*l + *ti*l ≥ *di*l ∨ *ti*l + *tk*l ≥ *dk*l

*tk*l ≥ *rk*

En donde:

*tn*: Tiempo de inicio del nodo sumidero, tiempo total o makespan del proceso.

*tkm*: Tiempo de inicio de la tarea k en la m´aquina *m*.

*dkl*: Duraci´on de la tarea k en la m´aquina *l*.

*rk*: Tiempo de liberaci´on de la tarea k [3].

**C´**

**alculo del Makespan en el Job Shop (JSSP)**

Para el problema del JSSP se pueden tener diferentes criterios de optimizaci´on como minimizaci´on del makespan, minimizaci´on de la tardanza promedio o maximizaci´on de las ´ordenes entregadas a tiempo. Generalmente estos objetivos se contraponen por lo que es recomendable estudiar a fondo el problema a resolver e identiﬁcar el criterio que se quiere optimizar. Se indica a continuaci´on la forma de calcular el makespan para el posterior desarrollo experimental del trabajo. El makespan es el mayor tiempo de proceso de todas las tareas a realizar siguiendo un secuenciamiento

***ICM 1-18 ESPOL***

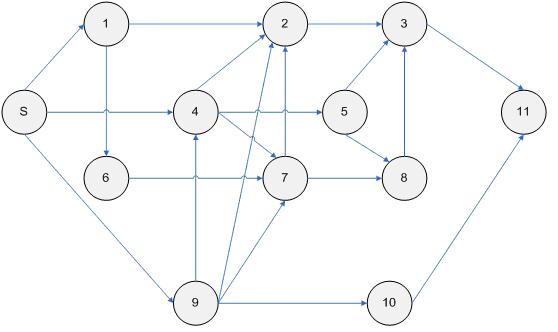
factible, para entender mejor el c´alculo se presenta un ejemplo num´erico: se cuenta con 4 tareas y 4 m´aquinas. En el siguiente cuadro se indica el n´umero de operaciones y los tiempos invertidos por tarea (para el efecto se ha numerado secuencialmente cada operaci´on).

Tabla 1.2: *Dato*s *par*a *c´alcul*o *de*l *makespa*n

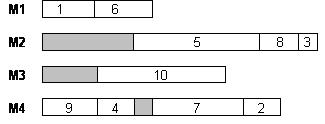
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| di | 2 | 2 | 1 | 2 | 7 | 4 | 5 | 2 | 3 | 7 |
| J(i) | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 |
| u(i) | M1 | M4 | M2 | M4 | M2 | M1 | M4 | M2 | M4 | M3 |

I Operaci´on  
Di Tiempo de proceso de la operaci´on i  
J(i) Tarea a la que pertenece la operaci´on i  
u(i) M´aquina donde se procesa la operaci´on i

Se presenta a continuaci´on la Figura 1.2 que contiene un grafo disyuntivo donde se presenta una de las soluciones factibles.



En la Figura 1.3 se aprecia una propuesta de secuenciamiento.



La ruta m´as larga para el c´alculo del makespan es cumplir las operaciones 94-5-8-3 lo que da un tiempo total de 15. El makespan obtenido en el ejemplo es el camino en el que una operaci´on es inmediatamente continuada con la siguiente operaci´on en la misma m´aquina o por la siguiente operaci´on de la tarea en una m´aquina diferente. El problema de obtener el menor makespan se reduce a obtener una selecci´on de arcos disjuntos que minimicen la longitud del camino m´as largo.

**1.4.3. El problema del secuenciamiento del Open Shop (OSSP)**

El Open Shop Scheduling Problem es un problema de secuenciamiento de la producci´on, que de forma general se describe as´ı:

En el OSSP no se tiene una secuencia predeﬁnida de operaciones entre las tareas y por lo tanto el OSSP tiene un espacio de soluciones mucho mayor al JSSP con dimensiones similares n × *m*. Probablemente, el mejor ejemplo de un open shop es un taller de reparaci´on de autos donde la secuencia de operaci´on/reparaci´on no est´a deﬁnida estrictamente. En general, en el OSSP, cada tarea debe ser procesada en cada m´aquina y consiste en una serie de mi operaciones las cuales tienen que ser secuenciadas en cualquier orden. De forma similar al FSSP y al JSSP cada tarea puede ser procesada en una m´aquina unicamente a la vez.

´

**1.5. Complejidad computacional de los problemas de secuenciamiento de la producci ´**

**on**

La complejidad computacional estudia todos los recursos requeridos durante el procesamiento de un algoritmo usado para resolver un problema determinado; en ´este caso un problema de secuenciamiento de la producci´on. Los recursos m´as analizados

***ICM 1-20 ESPOL***

son:

Tiempo: relacionado con el n´umero y tipo de pasos para la ejecuci´on del algoritmo.

Espacio: relacionado con la cantidad de memoria usada para la ejecuci´on del algoritmo.

De acuerdo al orden de complejidad los problemas se clasiﬁcan en:

Clase P: Los algoritmos de complejidad polin´omica se dice que son tratables en el sentido de que suelen ser abordables en la pr´actica.

Clase NP: Son problemas complejos intratables, los cuales se caracterizan por el hecho de que puede aplicarse un algoritmo polin´omico para comprobar si una posible soluci´on es v´alida o no. Esta caracter´ıstica lleva a un m´etodo de resoluci´on no determinista, que consiste en aplicar heur´ısticos para obtener soluciones hipot´eticas que se van desestimando (o aceptando) a ritmo polin´omico. Los problemas de esta clase se denominan NP (la N de no-deterministas y la P de polin´omicos).

Clase NP Completos: Se conoce una amplia variedad de problemas de tipo NP, de los cuales destacan algunos de ellos de extrema complejidad. Estos problemas se caracterizan por ser todos ¨ıguales.en el sentido de que si se descubriera una soluci´on P para alguno de ellos, esta soluci´on ser´ıa f´acilmente aplicable a todos ellos. Realmente, tras a˜nos de b´usqueda exhaustiva de dicha soluci´on, es un hecho aceptado que no debe existir, aunque nadie ha demostrado, todav´ıa, la imposibilidad de su existencia. Una alternativa para resolver los problemas NP-completos son los algoritmos heur´ısticos que si bien no encuentran la soluci´on ´optima a un problema NP si determinan buenas soluciones. Ejemplos de problemas NP-completos son el problema del agente viajero (TSP), el problema de coloreamiento de un grafo y el problema de la satisfacibilidad.

Clase NP Duros: Esta clase la cual es la m´as compleja puede ser descrita como aquella que contiene los problemas de decisi´on que son al menos tan dif´ıciles como un problema NP, sin embargo para los problemas NP-duros no

***ICM 1-21 ESPOL***

es posible encontrar un algoritmo que realice una b´usqueda exhaustiva. Cabe indicar que todos los problemas reales”de secuenciamiento de la producci´on son de clase NP-duros (para problemas cuyo n´umero de m´aquinas es mayor a tres) debido a que toman en cuenta muchos factores de un proceso sist´emico.

**1.6. Metodolog´ıas de soluci ´on para los problemas de secuen**

**ciamiento de la producci ´**

**on**

Los problemas de secuenciamiento de la producci´on son problemas complejos y sist´emicos donde las soluciones son representadas mediante variables discretas. Estos problemas pueden ser resueltos de dos formas:

Mediante algoritmos exactos o completos para problemas de menor complejidad (te´oricos y m´aximo 3 m´aquinas).

Mediante algoritmos de aproximaci´on para problemas m´as complejos y que consisten en encontrar soluciones cercanas a la ´optima. Se puede usar algoritmos heur´ısticos o meta-heur´ısticos.

**1.6.1. Algoritmos Exactos**

Los algoritmos exactos garantizan el poder encontrar la soluci´on optima en un ´tiempo de procesamiento razonable. Lamentablemente para los problemas NP-duros como los de secuenciamiento de producci´on no existe ning´un algoritmo exacto para resolverlos en un tiempo polinomial. Para problemas complejos de la vida pr´actica los algoritmos exactos necesitan tiempos computacionales exponenciales lo que lleva a soluciones no pr´acticas. La familia de los m´etodos exactos es considerablemente grande pero los m´etodos exactos m´as comunes para los problemas de secuenciamiento de la producci´on son los algoritmos branch and bound, programaci´on entera mixta y m´etodos de descomposici´on.

**1.6.2. Algoritmos de Aproximaci ´on**

“Debido a los altos tiempos de c´omputo necesarios para resolver algoritmos exactos en el secuenciamiento de la producci´on los investigadores se centraron

en m´etodos de aproximaci´on. En estos m´etodos se sacriﬁca la garant´ıa de encontrar una soluci´on ´optima en funci´on de obtener soluciones cercanas a la ´optima en tiempos computacionales razonables y pr´acticos. La forma b´asica de los algoritmos de aproximaci´on es llamada ”heur´ıstica”. La clasiﬁcaci´on m´as b´asica de los m´etodos heur´ısticos son: m´etodos constructivos y m´etodos de b´usqueda local”4 .

**Algoritmos Constructivos**

Los algoritmos constructivos generan soluciones mediante la adici´on gradual de partes a la soluci´on inicial (la cual es el conjunto vac´ıo). En los problemas de secuenciamiento de la producci´on, estas partes son operaciones. Las heur´ısticas constructivas son generalmente los algoritmos de aproximaci´on m´as r´apidos aunque con algunas implementaciones especiales aumenta el tiempo de procesamiento computacional. La gran ventaja en requerimientos de tiempo computacional est´a compensada con el hecho que generalmente obtiene soluciones de menor calidad cuando se las compara con las t´ecnicas de b´usqueda local. Entre las heur´ısticas constructivas m´as usadas en el secuenciamiento de la producci´on est´an las variantes de las reglas de despacho”. En la tabla 1.3 se encuentran las m´as comunes.

Debido a la facilidad de la implementaci´on y los bajos requerimientos computa-

Tabla 1.3: *Regla*s *d*e *despach*o *m´a*s *comune*s

|  |  |
| --- | --- |
| Regla | Descripci´on |
| SOT | Una operaci´on con el menor tiempo de proceso en la m´aquina considerada |
| LOT | Una operaci´on con el mayor tiempo de proceso en la m´aquina considerada |
| LRPT | Una operaci´on con el mayor tiempo de procesamiento de las tareas remantes |
| SRPT | Una operaci´on con el menor tiempo de procesamiento de las tareas remantes |
| LORPT | Una operaci´on con la cola m´as alta y menor tiempo del proceso |
| Random | La operaci´on para la m´aquina considerada es aleatoriamente escogida |
| FCFS | La primera operaci´on en la cola de trabajos esperados para la misma m´aquina |
| SPT | Un trabajo con el menor tiempo total de proceso |
| LPT | Un trabajo con el mayor tiempo total de proceso |
| LOS | Una operaci´on con el mayor tiempo de proceso de operaci´on posterior |
| SNRO | Una operaci´on con el menor n´umero de operaciones posteriores |
| LNRO | Una operaci´on con el mayor n´umero de operaciones posteriores |

cionales las reglas de despacho son las heur´ısticas m´as comunes para los problemas de secuenciamiento de la producci´on. Aunque se comportan bien en ciertos casos no existe una regla que se ajuste bien a todos los problemas de secuenciamiento; es

4Fatos Xhafa and Ajith Abraham. Metaheuristics for Scheduling in Industrial and Manufacturing Applications. 12 p.

***ICM 1-23 ESPOL on Log´ıstica***

decir que algunas de las reglas de despacho generan soluciones ´

optimas solamente en ciertos problemas.

**M´usqueda Local**

**etodos de B ´**

Los algoritmos de b´usqueda local parten con una soluci´on inicial (la mayor´ıa de veces generada por una heur´ıstica constructiva o aleatoria) e iterativamente tratan de reemplazar parte de la soluci´on con una mejor en un conjunto deﬁnido de soluciones del vecindario de inter´es. Con la ﬁnalidad de reemplazar partes de la soluci´on inicial los m´etodos de b´usqueda local desarrollan una serie de movimientos para formar nuevas soluciones en el mismo vecindario. Los movimientos m´as comunes son los siguientes: 2-Opt (act´ua sobre una serie de tareas de longitud al azar en una m´aquina), el cambio 1 − 1 (intercambia dos tareas de una misma m´aquina) y el cambio 1 − 0 (transﬁere una tarea de su posici´on en una m´aquina a otra posici´on en la misma m´aquina). Por supuesto el n´umero de posibles movimientos y los vecindarios correspondientes son pr´acticamente inﬁnitos. El principal inconveniente que presentan los m´etodos de b´usqueda local es que f´acilmente se quedan atrapados en ´usqueda. M´´

optimos locales debido a su naturaleza de b´as especıﬁcamente en la b´usqueda local realizando los movimientos apropiados se puede obtener un m´etodo muy efectivo en la exploraci´on del vecindario a partir de una soluci´on inicial pero no existe ning´un mecanismo que lleve a buscar mejores soluciones en otros vecindarios. Para remediar este problema se han desarrollado los m´etodos de b´usqueda local modernos (b´usqueda local modiﬁcada o VNS) con estrategias adicionales para buscar soluciones en otros vecindarios.

**1.6.3. Metaher´ısticas**

Debido al arduo inter´es de los investigadores se cre´o una nueva familia de algoritmos de aproximaci´on, la cual se ha constituido por las enormes ventajas que presenta, en la mejor forma de resolver los problemas de optimizaci´on combinatoria y especialmente los problemas de secuenciamiento de la producci´on. Este nuevo tipo de algoritmos b´asicamente combina los m´etodos heur´ısticos de una forma mejor estructurada. El principal objetivo de la nueva metodolog´ıa es explorar de una forma m´as efectiva y eﬁciente el espacio de b´usqueda basado en movimien-

***ICM 1-24 ESPOL***

tos l´ogicos y conocimiento del efecto de estos movimientos de forma que facilita el escape de los ´etodos metaheurısticos son mucho mejores

optimos locales. Estos m´´que los heur´ısticos principalmente en que generan soluciones robustas; sin embargo requieren mayores recursos computacionales, informaci´on del problema y son m´as dif´ıciles de implementar.

“Las metaher´ısticas son estrategias de un nivel avanzado las cuales permiten tener un mejor desempe˜no que las heur´ısticas tradicionales. El principal objetivo es evitar las desventajas del desarrollo iterativo y, en particular, el profundo descenso permitiendo a la b´usqueda local escapar de los ´optimos locales. Esto se puede conseguir permitiendo movimientos que llevan a soluciones de menor calidad o generando nuevas soluciones iniciales para la b´usqueda local de forma m´as inteligente que dando soluciones iniciales al azar. Muchos de los m´etodos pueden ser interpretados como la introducci´on de un bias de forma que se obtienen r´apidamente soluciones de alta calidad. Este bias de puede encontrar de varias formas y se clasiﬁca en bias descendente (basado en la funci´on objtivo), bias de memoria (basado en las decisiones previas) o bias de experiencia (basado en el desempe˜no primario). Muchas de las metaheur´ısticas usan decisiones probabilisticas durante la b´usqueda. Pero la principal diferencia con la b´usqueda al azar es que la aleatoriedad en los algoritmos metaheur´ısticos es usada de forma inteligente”5 .

Las metaheur´ısticas son estrategias que gu´ıan el proceso de b´usqueda. Los algoritmos exactos son demasiado lentos incluso cuando exploran espacios de soluciones muy peque˜nos, por otro lado las heur´ısticas realizan b´usquedas ciegas y no proporcionan soluciones robustas que puedan ser explicadas de forma l´ogica.

Recientes investigaciones han demostrado que los mejores m´etodos para los problemas de secuenciamiento de la producci´on son los que combinan t´ecnicas de b´usqueda local con metaheur´´

ısticas de forma que se permita el escape de los optimos locales aceptando movimientos que empeoran la soluci´on y de esta forma soluciones de menor calidad. Aunque el permitir movimientos que degradan la calidad de la soluci´on perecer´ıa contradictorio, estos movimientos apoyan a las metaheur´ısticas para escapar de los ´optimos locales con la esperanza que las nuevas soluciones inferiores est´en en la vecindad de un ´optimo global y por ende, una t´ecnica de b´usqueda local en esta vecindad de c´omo resultado una soluci´on ´optima global. Las estrategias usadas para escapar de los ´

optimos locales pueden dividirse en estrategias de diversiﬁcaci´on (exploraci´on efectiva de todos los vecindarios posibles del espacio de

5Blum C. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. 270 p.

***ICM 1-25 ESPOL***

soluciones) y de intensiﬁcaci´on (basados en un conocimiento particular que permite explorar una subespacio de soluciones m´as peque˜no).

**Clasiﬁcaci´on de las Metaher´ısticas**

La clasiﬁcaci´on m´as usada se basa en la cantidad de soluciones con las que puede trabajar cada metaheur´ıstica. En base a esto se clasiﬁcan en:

B´usqueda de punto simple: mejoran una soluci´on espec´ıﬁca explorando su vecindario con una serie de movimientos determinados.

Basados en poblaci´on: combinan un n´umero determinado de soluciones con la ﬁnalidad de generar nuevas soluciones que combinen las mejores caracter´ısticas de las antiguas y que sean de mejor calidad. Estos son generalmente m´etodos iterativos que reemplazan gradualmente las soluciones anteriores con las nuevas mejoradas.

Pr´acticamente todas las metaheur´ısticas se pueden aplicar para la resoluci´on de problemas de secuenciamiento de la producci´on. Sin embargo los m´etodos m´as usados por la efectividad en este tipo de problemas son:

Algoritmos evolutivos: Este tipo de algoritmos se dividen principalmente en 3 categor´ıas: algoritmos gen´eticos, estrategias evolutivas y programaci´on evolutiva. Uno de los algoritmos mas recientes en esta categor´ıa es el algoritmo evolutivo diferencial (DEA).

Colonia de hormigas: Se inspira en el comportamiento que rige a las hormigas de diversas especies para encontrar los caminos m´as cortos entre las fuentes de comida y el hormiguero.

B´usqueda dispersa (Scatter Search): Similar a los algoritmos gen´eticos pero usan estrategias sistem´aticas en lugar de aleatorias

Redes Neuronales: Constituyen una tecnolog´ıa de inteligencia artiﬁcial avanzada que simula el aprendizaje del cerebro y los procesos de toma de decisiones.

***ICM 1-26 ESPOL***

B´usqueda Local B´asica: Explora una vecindad de una soluci´on con un conjunto de movimientos con la ﬁnalidad de obtener un ´optimo local.

B´usqueda Local Explorativa: Puede ser GRASP (greedy randomized adaptative search procedure), VNS (variable neighboorhood search) y ILS (iterated local search).

Recocido Simulado: Basado en los cambios energ´eticos de un sistema de part´ıculas conforme disminuye la temperatura hasta que llega a un estado estable.

B´usqueda Tab´u: Utiliza un procedimiento de b´usqueda local para moverse iterativamente buscando una mejor soluci´on hasta satisfacer un criterio determinado. El procedimiento se fundamenta en una lista tab´u.

Aceptaci´on de umbral: Una modiﬁcaci´on de la b´usqueda local para permitir aceptar soluciones de menor calidad pero escapar de ´optimos locales.

**1.6.4. M´etodos H´ıbridos**

En los a˜nos recientes los investigadores se han concentrado en desarrollar m´etodos que combinan las fortalezas de las metaheur´ısticas y reducen sus debilidades. A estos m´etodos se les conoce como metaheur´ısticas h´ıbridas. Debido a estas caracter´ısticas es posible encontrar mejores soluciones con ´estos m´etodos. Se puede distinguir tres formas de hibridizaci´on:

Cambio de Componentes: Combinaci´on de metaheur´ısticas con t´ecnicas de b´usqueda local.

B´usqueda cooperativa: Intercambio de informaci´on entre dos o m´as metaheur´ısticas.

M´etodos de Integraci´on: Combina metaheur´ısticas y programaci´on con restricciones. Dependiendo del tipo de problema a resolver debe escogerse una de las tres alternativas.

***ICM 1-27 ESPOL on Log´ıstica***

**1.6.5. Metaheur´ısticas para el Job Shop Scheduling Problem**

Para resolver el JSSP se han usado varias metaheur´ısticas, como los algoritmos gen´eticos. Seg´un los investigadores ´este m´etodo no proporciona soluciones cercanas a la ´optima. Es decir los algoritmos gen´eticos fallan en la b´usqueda de mejores regiones de una vecindad. El uso de colonia de hormigas para resolver este problema es muy limitado ya que el m´etodo funciona mejor con variables continuas por lo que generalmente se lo usa combinado con otras metaheur´ısticas. Los mejores m´etodos en la resoluci´on de estos problemas se encuentran con el uso de b´usqueda variable local (VNS). Sin embargo en t´erminos de aplicaciones pr´acticas los m´etodos que dan mejores soluciones son muy complejos de implementar y requieren muchos recursos computacionales por lo que el uso de algoritmos gen´eticos los cuales son muy simples en comparaci´on con los otros m´etodos es una buena alternativa.

**1.7. Algoritmos Gen´eticos**

Los algoritmos gen´eticos propuestos por Hollan en 1975 son metaheur´ısticas que usan estrategias de b´usqueda general que permiten resolver problemas de naturaleza combinatoria. Los algoritmos gen´eticos simulan el proceso de evoluci´on de los organismos vivos y est´an basados en el principio de *supervivenci*a *de*l *m´a*s *apt*o para formar la siguiente generaci´on de posibles soluciones. Estas soluciones se construyen utilizando operadores probabil´ısticos, como el cruce gen´etico, mutaci´on y supervivencia del m´as apto (selecci´on). A medida que avanza el proceso evolutivo los individuos m´as aptos sobreviven, lo que representa las mejores soluciones, mientras que los individuos menos aptos desaparecen. La evoluci´on de dichas soluciones hacia valores ´en buena medida de una

optimos del problema depende adecuada codiﬁcaci´on de ´estas.

**1.7.1. Espacio de B´usqueda en Algoritmos Gen´eticos**

El espacio de b´usqueda es el conjunto de todas las posibles soluciones de un problema concreto. El espacio de b´usqueda puede ser bien deﬁnido, pero en la mayor´ıa de las ocasiones s´olo se conocen algunos puntos. A cada posible soluci´on se le puede asociar una aptitud o puntuaci´on que representa la bondad de ´esta solu-

***ICM 1-28 ESPOL***

ci´on para el problema. En la naturaleza esto equivaldr´a al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptaci´on de un individuo al problema, mayor ser´a la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material gen´etico con otro individuo seleccionado de igual forma. Este cruce producir´a nuevos individuos descendientes de los anteriores (los cuales comparten las mejores caracter´ısticas de sus padres) y que reemplazan la soluci´on anterior. Un algoritmo gen´etico devolver´a la mejor soluci´on de entre todas las posibles que tenga en un momento dado. Cuando se usa un algoritmo gen´etico las posibles soluciones generan otras soluciones a medida que el algoritmo evoluciona. Las caracter´ısticas de los algoritmos gen´eticos son:

Operan en una poblaci´on y no en puntos aislados.

Operan en un espacio de soluciones codiﬁcadas y no directamente en el espacio de b´usqueda.

Solamente requiere informaci´on del valor de la funci´on objetivo de cada individuo

Usa transiciones probabil´ısticas y no determin´ısticas.

**1.7.2. Conceptos B´eticos**

**asicos de los algoritmos gen´**

Individuo o cromosoma: Un individuo determina una posible soluci´on del problema que se pretende resolver mediante el algoritmo gen´etico. Las variables de decisi´on a ser optimizadas generalmente se representan como un vector. Un cromosoma se compone de un n´umero determinado de genes, los cuales pueden ser representados por un alfabeto binario o uno de mayor cardinalidad.

Poblaci´on: Conjunto de individuos con los que se trabaja en el algoritmo gen´etico. En un algoritmo gen´etico los individuos que constituyen la poblaci´on van cambiando pero generalmente el tama˜no de la misma permanece constante.

Funci´on de ajuste: Se trata de una funci´on evaluadora de la calidad de un individuo como soluci´on al problema. Permite ordenar a los individuos de la poblaci´on de acuerdo a la bondad de los mismos.

***ICM 1-29 ESPOL***

Cruce: Es una de las operaciones fundamentales que intervienen en todo algoritmo gen´etico. Como norma general se aplica despu´es de un proceso de selecci´on de dos individuos y consiste en una combinaci´on de los mismos para obtener como resultado otros dos nuevos individuos.

Mutaci´on: Constituye otra operaci´on fundamental en un algoritmo gen´etico. En este caso se selecciona un individuo, el cual sufre una peque˜na modiﬁcaci´on aleatoria en su codiﬁcaci´on obteni´endose otro individuo nuevo.

**1.7.3. Estructura b´eticoasica de un algoritmo gen´**

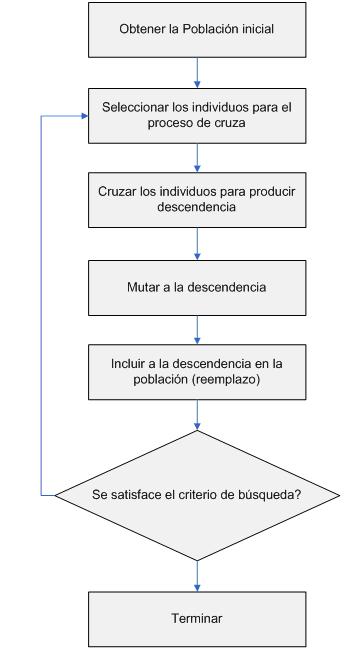
La ejecuci´on b´asica de un algoritmo gen´etico se presenta en la Figura 1.4. El proceso empieza con la obtenci´on de la poblaci´on inicial ya sea mediante una heur´ıstica

o aleatoriamente. Luego se selecciona los individuos para el cruce en funci´on del valor de la funci´on objetivo, para esto se calcula el valor de la funci´on objetivo de todos los individuos. En ´esta fase los individuos con bajos valores de la funci´on objetivo son eliminados y reemplazados por individuos *m´a*s *apto*s de forma que la poblaci´on siempre conserva el tama˜no original. El siguiente paso consiste en realizar el cruce entre los individuos (los m´as aptos tienen m´as probabilidad de ser seleccionados para el cruce). Posteriormente se realiza una mutaci´on (alteraci´on aleatoria de un gen de un individuo o de la poblaci´on) a los cromosomas de acuerdo a una probabilidad determinada. Al igual que en la naturaleza la mutaci´on tiene una peque˜na probabilidad de ocurrencia y afecta ´unicamente un peque˜no n´umero de individuos. Luego de la mutaci´on se tiene la siguiente poblaci´on del mismo tama˜no que la inicial y se veriﬁca el cumplimiento del criterio de b´usqueda.

**1.7.4. Par´eticosametros de los algoritmos gen´**

Porcentaje de Cruce (Pc): Indica con que frecuencia se cruzar´an los individuos. Si este es 0 %, los hijos ser´an como los padres y s´olo ser´an alterados por la mutaci´on. Si ´este es 100 % todos los individuos nuevos ser´an creados mediante cruce de los padres de la generaci´on previa. Cuanto m´as se crucen los individuos se supone que los hijos ser´an mejores. Sin embargo es recomendable, que algunos individuos pasen sin modiﬁcar a la siguiente generaci´on. Si

***ICM 1-30 ESPOL***



el porcentaje de cruce es muy bajo la convergencia ser´a muy lenta y se desperdiciar´a tiempo hasta llegar a la soluci´on, por otro lado si el porcentaje de cruce es muy alto se puede converger en una mala soluci´on de forma temprana. Dependiendo del problema puede estar ente 60 % a 95 %.

Porcentaje de Mutaci´on (Pm): Establece la probabilidad con la cual los individuos ser´an mutados. Si ´este porcentaje es 0 % los individuos generados despu´es de aplicarse el cruce no sufrir´an ning´un cambio por el contrario si es de 100 % todos lo individuos de la poblaci´on sufrir´an cambios o mutaciones. La mutaci´on trata de impedir que la b´usqueda del gen´etico caiga en ´optimos locales por eso es conveniente que ocurra ocasionalmente. No es bueno, sin embargo, que la mutaci´on ocurra continuamente, ya que la b´usqueda del

***ICM 1-31 ESPOL***

gen´etico pasa de ser *inteligent*e a b´usqueda aleatoria. El porcentaje de mutaci´on generalmente toma valores cercanos a L 1 , donde L es la longitud del cromosoma.

Tama˜no de la Poblacion: Establece cu´antos individuos habr´a en cada una de las generaciones. Si el tama˜no de la poblaci´on es muy bajo, el algoritmo gen´etico tiene pocas posibilidades de evolucionar por el cruce y los individuos nuevos se parecer´an mucho a sus padres. Tampoco un tama˜no excesivo es adecuado porque se llega a un punto en el que los resultados no mejoran. Lo ideal es, en funci´on del problema y la codiﬁcaci´on, establecer un l´ımite adecuado del tama˜no de la poblaci´on.

N´umero de Generaciones: Es el n´umero de iteraciones del proceso, con el paso de las generaciones la poblaci´on del algoritmo gen´etico evolucionar´a obteniendo cada vez mejores individuos. Conviene, al igual que con el tama˜no de la poblaci´on, ﬁjar un n´umero de generaciones adecuado para conseguir el resultado deseado.

Tama˜no del Individuo: Depender´a del n´umero de elementos que constituyan una soluci´on. Var´ıa con el problema a resolver.

**1.7.5. Procesos de un algoritmo gen´etico**

Codiﬁcaci´on del Problema: Es esencial distinguir en una soluci´on a un problema entre el genotipo y el fenotipo. El genotipo es la representaci´on interna que se utiliza para trabajar con la soluci´on, mientras que el fenotipo es la soluci´on en s´ı misma. Los operadores del algoritmo trabajar´an con el genotipo. Existen algunas formas de representar el genotipo:

Representaci´on Binaria: La representaci´on binaria es la m´as com´un. En ella, un cromosoma es una cadena de bits 0 ´

o 1. Las primeras investigaciones en gen´eticos utilizaron este tipo de codiﬁcaci´on debido a su sencillez. Es una representaci´on indirecta.

Representaci´on basada en Permutaciones: Una permutaci´on de los elementos de un determinado conjunto Z induce un orden total en ellos.

***ICM 1-32 ESPOL***

Tabla 1.4: *Representaci´o*n *Binari*a



Esta relaci´on de orden total puede ser representada mediante una matriz de 1 y 0 llamada matriz de precedencia. El elemento de la matriz a*i*j colocado en la ﬁla i y la columna j es 1 (Verdadero) si y solo si el s´ımbolo etiquetado como i antecede al s´ımbolo etiquetado como j en la secuencia. Esto da lugar a una primera forma de manipular una permutaci´on, como una colecci´on no ordenada de relaciones de precedencia. En ciertas ocasiones esta representaci´on es demasiado exhaustiva y conviene relajarla un poco utilizando el concepto de adyacencia. En esta representaci´on una permutaci´on es un vector de | Z | (cardinal del conjunto que estamos considerando) pares, cada uno de ellos indicando cual es el inmediato predecesor de cada uno de ellos. La forma mas natural para representar una permutaci´on es la basada en la posici´on, es decir una lista ordenada de los elementos del conjunto Z. Tambi´en se puede considerar una representaci´on basada en bloques, siendo un bloque un subconjunto de elementos contiguos. De todas estas posibles representaciones se elegir´a la basada en la posici´on por considerarse mas sencilla de usar y suﬁcientemente eﬁcaz para el problema a resolver. Es una representaci´on indirecta.

Tabla 1.5: *Representaci´o*n *basad*a *e*n *Permutacione*s



Representaci´on Directa: En este tipo de representaci´on se usa directamente en problemas que utilizan valores dif´ıciles de representar. Por ejemplo si se usan n´umero reales. Se trata de problemas en los que una codiﬁcaci´on binaria ser´ıa demasiado complicada. Cada cromosoma es una secuencia de valores. Los valores son cualquier cosa relacionada con el tipo problema. (N´umeros reales, caracteres u otros tipos de objetos).

Tabla 1.6: *Representaci´o*n *Direct*a

|  |  |
| --- | --- |
| CROMOSOMA A | 1.2324 5.3243 0.4556 2.3293 2.4545 |
| CROMOSOMA B | ABDJEIFJDHDIERJFDLDFLFEGT |
| CROMOSOMA C | (atr´as), (atr´as), (derecha), (hacia delante), (izquierda) |

Selecci´on: Es el proceso de escoger dos padres de la poblaci´on para cruzarse. Luego de que se ha escogido una codiﬁcaci´on adecuada para el problema es vital decidir la forma de escoger a los individuos que crearan descendencia para la siguiente generaci´on. De acuerdo con la teor´ıa de la evoluci´on de Darwin, s´olo los mejores individuos se cruzan, es decir el prop´osito de la selecci´on es enfocarse en los mejores individuos con la esperanza que su descendencia tenga una mejor adaptaci´on. Bas´andose en esto existen varios m´etodos que son utilizados por los algoritmos gen´eticos sin embargo pueden clasiﬁcarse en dos tipos la selecci´on proporcional y la selecci´on basada en orden. En la selecci´on proporcional se eligen individuos en funci´on de sus valores relativos de ajuste respecto al de otros individuos en la poblaci´on. Por otro lado en la selecci´on basada en orden se toma en cuenta un ranking para los individuos. La selecci´on debe ser equilibrada con el cruce y la mutaci´on debido a que una selecci´on muy rigurosa puede hacer que se pierda diversidad de los individuos. Los m´etodos m´as conocidos de selecci´on son:

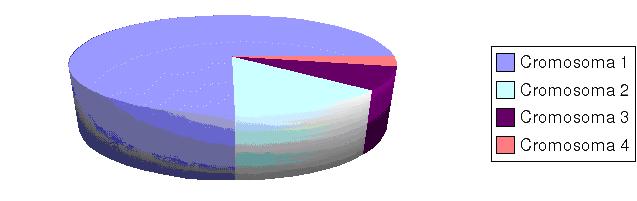
Selecci´on por la Regla de la Ruleta: Los padres se seleccionan de acuerdo a su adaptaci´on. Los mejores individuos son los que tienen may-ores posibilidades de ser elegidos. Intuitivamente el proceso construye una ruleta en la que cada uno de las porciones representa a un individuo. El tama˜no de cada porci´on es proporcional a la adaptaci´on del individuo, as´ı los individuos buenos se llevar´an las mayores porciones y al rev´es ocurrir´a con los peores. A continuaci´on se describe el proceso de selecci´on por ruleta:

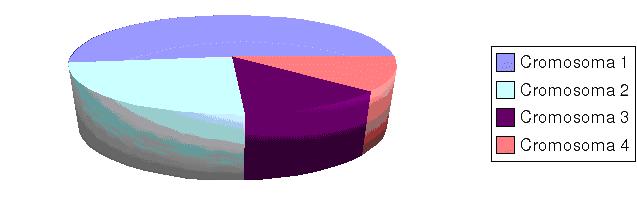


Ahora, se gira la ruleta N veces, donde N es el n´umero de individuos de

la poblaci´on. Existe un algoritmo para realizar este proceso: Calcular la suma total acumulada de la funci´on de evaluaci´on de todos los individuos de la poblaci´on actual.

Generar un n´umero aleatorio entre 0 y la suma total. Recorrer la poblaci´on sumando los valores de la funci´on de ajuste hasta que la suma sea mayor o igual a *r*(probabilidad elegida), en ´este punto se selecciona el individuo que aport´o al ﬁnal para que se supere o iguale el valor de *r*. Este tipo de selecci´on depende de la variaci´on entre los valores de la funci´on de los individuos, cuando son muy diferentes no funciona bien. Por ejemplo si un cromosoma ocupa el 90 % de la ruleta el resto de los cromosomas tienen muy pocas posibilidades de ser elegidos. Selecci´on por Ranking: Los individuos son ordenados de acuerdo a su valor de la funci´on objetivo en un ranking, de esta manera si se tiene N cromosomas el individuo con peor ajuste se le asignar´a un 1 y el que tenga el mejor ajuste se le asignar´a la posici´on *N*. Con ´este m´etodo se tiene una convergencia m´as lenta pero previene de quedar atrapado en un ´optimo local. En la ﬁgura 1.6 y la ﬁgura 1.7 se indica c´omo cambia la situaci´on antes y despu´es del ranking.





Ahora todos los cromosomas tienen la oportunidad de ser seleccionados. Sin embargo este m´etodo puede hacer que el gen´etico converja lentamente

***ICM 1-35 ESPOL***

a la soluci´on, ya que los mejores individuos no presentan gran diferencia de los peores. A este esquema se le pueden aplicar otros criterios; un ejemplo es el muestreo estoc´astico universal, que trata de evitar que los individuos con m´as adaptaci´on copen la poblaci´on. En lugar de dar la vuelta a una ruleta con una ranura, se da la vuelta a la ruleta con N ranuras, tantas como la poblaci´on. De esta forma, la distribuci´on estad´ıstica de descendientes en la nueva poblaci´on es m´as parecida a la real.

Selecci´on por Torneo K/L: La selecci´on por Torneo K�Lconsiste en seleccionar K individuos de la poblaci´on aleatoriamente y de estos K individuos se seleccionan los L que tengan mejor adaptaci´on. Este proceso se repite todas las veces necesarias hasta formar la nueva poblaci´on. Este es uno de los m´etodos de selecci´on m´as utilizados actualmente. Se utiliza tambi´en en algunos algoritmos en el momento de la aceptaci´on.

Elitismo: Este concepto expresa la idea de que el mejor individuo de la actual generaci´on pase sin ser modiﬁcado a la siguiente generaci´on. De esta forma no se perder´a el mejor cromosoma. Al resto de la poblaci´on se le aplica la reproducci´on normalmente. Por otra parte existen algoritmos gen´eticos llamados elitistas debido a que convergen muy r´apidamente a la soluci´on.

Cruce: El cruce trata de crear una generaci´on de individuos nuevos (descendencia) pidiendo informaci´on a sus padres, por lo que luego del cruce la poblaci´on se enriquece con mejores individuos. Inicialmente puede parecer que el operador de cruce se limita a obtener informaci´on b´asica pero realmente se fundamente en otros factores, de los cuales depende el tipo de operador.

Cruce en un Punto: Usado en los algoritmos gen´eticos tradicionales. Se copian los genes del primer padre hasta el punto de corte y se rellena con el resto de elementos que hagan la soluci´on v´alida en el orden en que aparecen en el segundo padre considerando la cadena de genes como c´ıclica. En el caso de que se haya utilizado una codiﬁcaci´on binaria simplemente se copian el resto de genes del segundo padre. En la ﬁgura 1.8 se observa un ejemplo del cruce en un punto.

***ICM 1-36 ESPOL***

11001011+11011111 = 11001111

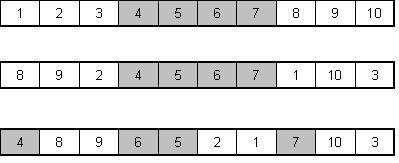


Cruce en dos Puntos: El a˜nadir puntos de cruce reduce el desempe˜no del algoritmo gen´etico debido a que se tiende a romper la cadena c´ıclica de los padres. Sin embargo la ventaja de adicionar puntos es que se explora el espacio del problema de forma m´as efectiva. Para realizar el cruce en dos puntos se copian los genes del primer padre comprendidos entre los dos puntos de cruce y se rellenan los que faltan con los del segundo padre considerando la cadena de genes como c´ıclica. En las ﬁguras 1.9 y 1.10 se observa un ejemplo del cruce en dos puntos.

11001011 + 11011111 = 11011111



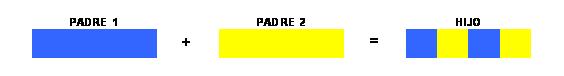
11001011 + 11011111 = 11011111



***´***

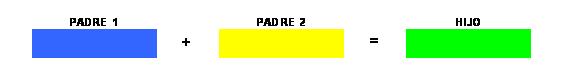
Cruce uniforme: Cada gene de la descendencia se crea mediante la copia del gen correspondiente de uno de los padres. Se escoge aleatoriamente si el gen i-´esimo del hijo se toma del primer o del segundo padre. En la ﬁgura 1.11 se observa un ejemplo del cruce uniforme.

11001011 + 11011101 = 11011111



Cruce aritm´etico: Se realizan operaciones aritm´eticas con los genes de los padres que dan como resultado la codiﬁcaci´on gen´etica del hijo. En la ﬁgura 1.12 se observa un ejemplo del cruce aritm´etico.

11001011 + 11011111 = 11001001



Cruce de tres padres: Se escogen aleatoriamente tres padres. Una parte del primer padre se compara con la misma parte del segundo padre. Si son iguales se elige esta parte para la descendencia, de lo contrario se toma la parte del tercer padre. En el cuadro 1.7 se observa un ejemplo del cruce de tres padres.

Tabla 1.7: *Cruc*e *d*e *tre*s *padre*s

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Padre 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| Padre 2 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| Padre 3 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| Hijo | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

Cruce de combinaci´on parcial (PMX o Partial Mapped Cross:

Este m´etodo de cruce fue propuesto por Goldberg y se aplic´o inicialmente

al problema del agente viajero En peste problema los cromosomas son secuencias de enteros; donde cada entero representa una ciudad diferente. El m´etodo propuesto garantiza que todas las ciudades est´en presentes solamente una vez en la descendencia, el procedimiento se tiene a continuaci´on:

Se elige aleatoriamente dos puntos de cruza.

Se intercambia estos dos segmentos en los hijos que se generan (igual

a la cruza de dos puntos).

Del resto de las cadenas que conforman los hijos se obtienen haciendo

mapeo entre los dos padres:

* Si un valor no est´a contenido en el segmento intercambiado permanece igual.
* Si est´a contenido en el segmento intercambiado, entonces se sustituye por el valor que tenga dicho segmento en el otro padre. En siguientes ﬁguras 1.13, refpmx1, refpmx2 y refpmx3 se observa un ejemplo.



Figura 1.13: *PMX*: *Lo*s *padre*s *so*n



Figura 1.14: PMX: Los hijos son

Para completar a los dos hijos se copia los valores que no est´an en el segmento intercambiado. Otro ejemplo es el siguiente: se tiene dos padres P1



Figura 1.15: PMX: Completar los hijos

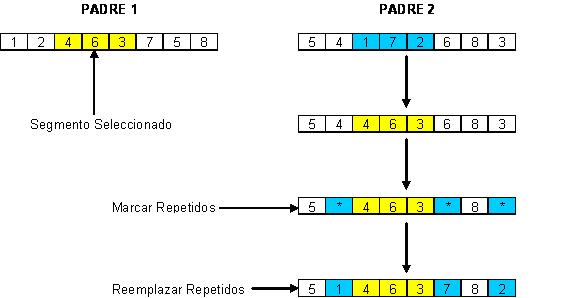
= (1,2,4,6,3,7,5,8) y P2 = (5,4,1,7,2,6,8,3) y el segmento seleccionado al azar de P1 para ser insertado en P2 es el (4,6,3) esto establece una relaci´on

***ICM 1-39 ESPOL***

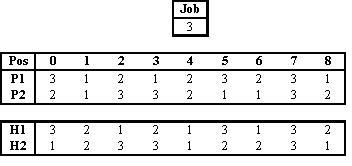


Figura 1.16: *PMX*: *S*e *mapea*n *lo*s *valore*s *restante*s

con el segmento (1,7,2) que ocupa las mismas posiciones en P2. Entonces la secuencia de operaciones transformar´ıan a P2 en (5,4,4,6,3,6,8,3) y luego eliminando las repeticiones quedar´ıa (5,\*,4,6,3,\*,8,\*) donde los asteriscos corresponden a los elementos repetidos que deben eliminarse de la conﬁguraci´on. Reemplazando queda (5,1,4,6,3,7,8,2), el n´umero 5 ocupa la posici´on del n´umero 1 por ´esta raz´on el primer asterisco es llenado con el n´umero 1, el n´umero 6 ocupa la posici´on del n´umero 7 y el ´ultimo asterisco es llenado con el n´umero 2. En la ﬁgura refpmx4 se muestra graﬁcamente explicado.



Cruce JOX (Job Order Cross: el operador de cruce jox permite trabajar sobre las representaciones de problemas que usan la codiﬁcaci´on de permutaciones. El operador de cruce genera dos hijos, para ello es necesario seleccionar de forma aleatoria una operaci´on de entre todas las existentes. La operaci´on o gen seleccionado mantendr´a su posici´on en los dos cromosomas hijos; y, los genes diferentes al seleccionado se intercambian entre los dos cromosomas padres para generar dos nuevos hijos.



Mutaci´on: La mutaci´on ocurre luego del cruce y tiene la ﬁnalidad de prevenir que el algoritmo se atrape en un ´on se encarga

optimo local. La mutaci´de mantener la diversidad gen´etica en la poblaci´on ya que introduce nuevas estructuras gen´eticas en la poblaci´on mediante la modiﬁcaci´on aleatoria de ciertos bloques. Existen diversas formas de mutaci´on, las cuales dependen de la codiﬁcaci´on elegida, las mismas que se describen a continuaci´on:

Inversi´on de genes: Se seleccionan genes aleatoriamente y se invierte su valor. Se utiliza en representaciones de bits, cambiando 0 por 1 o viceversa.

11001001 10001001

→



Cambio de orden: Se seleccionan dos genes aleatoriamente y se intercambian sus posiciones. Se utiliza en representaciones basadas en permutaciones.

(123456897) (183456297)

→

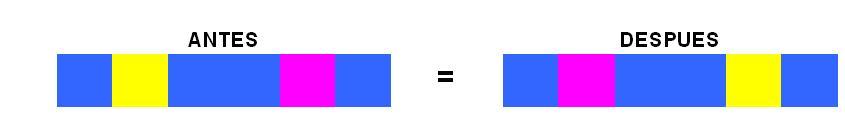


Figura 1.20: Cambio de Orden

Modiﬁcaci´on de Genes: Se realizan peque˜nas modiﬁcaciones en los genes. Por ejemplo en una codiﬁcaci´on basada en n´umeros reales se realizan sumas de n´umeros muy peque˜nos positivos o negativos.

(1*,*295*,*682*,*864*,*115*,*55) (1*,*295*,*682*,*734*,*225*,*55)

→

Aceptaci´on: Despu´es de realizar el cruce y la mutaci´on de los individuos de la poblaci´on se tiene que dos padres se cruzaron y se obtuvieron dos hijos, pero no todos pueden regresar a la poblaci´on original. El operador de aceptaci´on se encarga de decidir los miembros actuales, de llegar a haber alguno, que ser´an reemplazados por las nuevas soluciones. Existen algunos criterios de aceptaci´on:

Aceptaci´on total: Es la manera m´as utilizada, todos los hijos genera-dos son aceptados reemplazando a los padres y pasan a formar parte de la nueva poblaci´on.

De mejora: Los hijos pasan a la nueva poblaci´on si son mejores que los peores individuos de la poblaci´on actual.

Estado constante (Steady State): Se cambia un miembro de la poblaci´on a la vez. Para llevar a cabo la selecci´on se escoge un miembro de la poblaci´on de acuerdo a su funci´on de aptitud. Se copia ´este individuo y se procede a mutar la copia. Se selecciona un segundo miembro de la poblaci´on el cual es reemplazado por la cadena mutada.

Por torneo: De los cuatro individuos, dos hijos y dos padres, solo los dos m´as aptos, sean hijos o padres regresan a la poblaci´on.

**1.8. Programaci ´on en C++**

C++ es un lenguaje de programaci´on muy potente que fue creado por Bjarne Stroustrup en la d´ecada de los 80. Inicialmente fue una extensi´on del lenguaje C que se llam´o C *co*n *clases*. Inmediatamente adquiri´o popularidad ya que potenci´o la ﬂexibilidad y eﬁcacia que tenia el lenguaje C y se lo llam´o as´ı para referenciar el car´acter del operador de incremento de C (++). Actualmente el C++ es un lenguaje vers´atil, potente y general por lo que permite programar sistemas operativos,

***ICM 1-42 ESPOL***

compiladores, aplicaciones de bases de datos, procesadores de texto, juegos, entre otros.

Una de las ventajas del lenguaje C++ es que posee una gran cantidad de operadores lo cual le permite tener una expresividad notable cuando se implementan aplicaciones cient´ıﬁcas o matem´aticas; adicionalmente la sintaxis de clases y objetos permite manipular convenientemente diversas estructuras de datos y operaciones; las excepciones permiten procesar de un modo claro los casos de error.

C++ es un lenguaje consistente y sin ambig¨uedades pero si tiene desventajas como que deja algunos aspectos a criterio del implementador como el tama˜no del tipo de datos, por lo que se requiere de conocimientos medianos cuando se empieza a programar en este lenguaje.

Para facilitar la programaci´on y tener mayor facilidad para programar aplicaciones en Internet, se han creado programas como JAVA, PHP y .NET. Sin embargo, la potencia del Lenguaje C++ hace que hoy en d´ıa es uno de los lenguajes de programaci´on m´as usados en el mundo.

**CAPITULO 2**

**DESCRIPCION Y FORMULACION DEL PROBLEMA**

En la presente Tesis de Grado inicialmente se estudia el proceso de planiﬁcaci´on actual para luego proponer la optimizaci´on pertinente.

**2.1. M´on Actual etodo de Planiﬁcaci ´**

La Empresa ABC usa las siguientes fases en el m´etodo de planiﬁcaci´on actual-mente usado:

Generaci´on de un Plan Maestro de Producci´on (MPS) a partir del presupuesto comercial, este plan indica la cantidad de cada producto que debe ser fabricada semanalmente con un horizonte anual. Esta parte es realizada por el ERP que existe en la Empresa ABC y requiere de una aﬁnaci´on ﬁnal.

Semanalmente se realiza la planiﬁcaci´on ﬁnita con un horizonte de un mes y un per´ıodo congelado de dos semanas. Para ´esta planiﬁcaci´on ﬁnita se inicia con el MPS generado anteriormente y se distribuye los productos a generarse diariamente en funci´on de las restricciones de ingreso de materia prima. Una vez que se ha generado el plan diario el jefe de planiﬁcaci´on, jefe y gerente de producci´on generan manualmente la secuencia de producci´on y asignaci´on de mano de obra requerida para las tareas en funci´on de la experiencia sin tomar en cuenta ninguna t´ecnica de optimizaci´on formal. Toda ´esta labor

***ICM 2-44 ESPOL***

toma aproximadamente dos d´ıas semanalmente lo cual es un tiempo alto y ﬁnalmente no se obtiene un plan ´optimo de producci´on.

**2.2. Situaci ´on real de la Empresa ABC**

Como se indic´o anteriormente la tarea de planiﬁcaci´on ﬁnita en la Empresa ABC toma 2 d´ıas de tiempo del gerente de producci´on y jefes de planiﬁcaci´on y producci´on. Adicionalmente no se tiene una secuencia ´optima de producci´on, para conocer m´as a fondo la situaci´on actual se obtiene la siguiente informaci´on.

**2.2.1. Tiempo de Para en Celdas de Producci ´on**

En cada celda de producci´on se tiene informaci´on del tiempo diario improductivo y las razones para estos tiempos de paro. Se presenta a continuaci´on la informaci´on del tiempo de paro no programado con las causas de los ´

ultimos 3 meses.

Tabla 2.1: *Tiemp*o *d*e *par*a *promedi*o *diari*o

|  |  |
| --- | --- |
| Causa | Tiempo (min) |
| Secuenciaci´on de la producci´on | 121 |
| Da˜no de m´aquina | 44.4 |
| Falta de personal | 31.6 |
| Otros | 20.3 |
| Total | 217.3 |

Cabe indicar que para obtener esta informaci´on, se entreg´o a los operadores l´ıderes de cada c´elula un formato para registrar los tiempos y motivos de para. Posteriormente, el supervisor de producci´on tabulaba los datos a diario y atribu´ıa las causas reales. Se entiende como para de producci´on por mal secuenciamiento, a toda para de la c´elula de producci´on debida a que la operaci´on anterior no estaba completa. Se puede observar en la Tabla 4.6 que en la Empresa ABC se tienen serios problemas de productividad pues se trabajan diariamente dos turnos de 9 horas aproximadamente. Es decir 1080 minutos al d´ıa de los cuales las m´aquinas paran en promedio 217,3 minutos lo que equivale aproximadamente al 20,1 %. La Empresa ABC, por estos problemas de productividad incurre en costos innecesarios los que se detallan a continuaci´on:

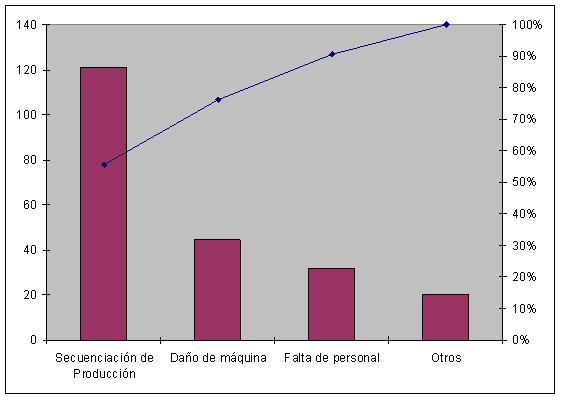
***ICM 2-45 ESPOL***

Pago de horas extras innecesarias al personal de producci´on, mantenimiento, calidad y bodega.

Pago de recorridos no programados para el personal.

Pago de servicios complementarios para producci´on como electricidad, vapor, agua y aire.

Estos gastos incrementan los costos de producci´on y le quitan una ventaja competitiva a la Empresa ABC. En la Figura 2.1, se presenta un diagrama de Pareto de los motivos de para con la ﬁnalidad de conocer la causa ra´ız de la p´erdida de productividad.



Gracias a la informaci´on recopilada para el presente proyecto de titulaci´on tambi´en se logra evidenciar que la Empresa ABC aparte de mejorar la planiﬁcaci´on ﬁnita de producci´on debe enfocarse adicionalmente en solucionar los frecuentes da˜nos de m´aquinas. Para esto se est´a trabajando en un Plan de mantenimiento Preventivo para lo cual se estableci´o rutas de inspecci´on mec´anica, tableros de alertas y se

***ICM 2-46 ESPOL***

compr´o equipo para realizar Mantenimiento Predictivo como: Medidor de vibraci´on, C´amara termogr´aﬁca y estraboscopio.

En la tabla 4.6 se presenta la informaci´on detallada de para en cada c´elula de producci´on debido a un mal secuenciamiento de las operaciones.

Tabla 2.2: *Tiemp*o *d*e *par*a *promedi*o *diari*o

|  |  |  |
| --- | --- | --- |
| Celda | Tiempo Promedio (min) | Causa |
| Pelado MP Picado MP Lavado MP Procesar MP Lavado Envases Preparaci´on LG Envasado Cerrado Pasteurizado Embalado | 7.2 11*,*1 8.1 35.7 0.8 18.7 6.2 5.7 23.1 4.4 | Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on Secuenciaci´on de la Producci´on |

**2.2.2. Horas extras pagadas no planiﬁcadas**

Debido a que los trabajos ya programados por d´ıa se cumplen al 100 % el indicador que reﬂeja la p´erdida de productividad por paras no programadas son las horas extras no planiﬁcadas que se pagan diariamente. Para calcular este indicador se procede de la siguiente forma:

Con el plan diario de producci´on (art´ıculos a fabricar y cantidades) se determina la cantidad de horas que se debe trabajar cada celda de producci´on, para ´este c´alculo se usan las rutas de producci´on las cuales proporcionan informaci´on del n´umero de personas y tiempo necesario por cada una de las operaciones.

Diariamente, mediante el reloj dactilar de la Empresa ABC, se registra el n´umero de horas que cada persona trabaja, ´este tiempo se almacena en el sistema inform´atico, el cual reconoce a que c´elula de producci´on pertenece dicho empleado. Finalmente con ´esta informaci´on se obtiene la diferencia entre las horas planiﬁcadas y las no programadas determinando as´ı el tiempo extra que fue requerido para cumplir la producci´on.

En la Tabla 2.4 se presenta el resumen a˜no de las horas extras no planiﬁcadas por mes desde noviembre de 2008 pues ´esta informaci´on esta disponible en la Empresa

***ICM 2-47 ESPOL***

ABC desde antes de la propuesta del presente proyecto.

Tabla 2.3: *Hora*s *extr*a *n*o *planiﬁcada*s *po*r *me*s

|  |  |  |  |
| --- | --- | --- | --- |
| Mes | Horas extras 50 % | Horas extras 100 % | Total |
| Noviembre | 3729 | 2128 | 5857 |
| Diciembre | 3508 | 2977 | 6485 |
| Enero | 2753 | 1920 | 4673 |
| Febrero | 2411 | 1754 | 4165 |
| Marzo | 2540 | 2022 | 4562 |
| Abril | 2016 | 1987 | 4003 |
| Mayo | 2870 | 1520 | 4390 |
| Junio | 2283 | 1948 | 4231 |
| Julio | 2229 | 2187 | 4416 |
| Agosto | 3498 | 1312 | 4810 |
| Septiembre | 3294 | 1829 | 5123 |
|  | TOTAL |  | 52715 |

Si se considera que en promedio una hora extra al 50 % cuesta $1,76 USD y una hora extra al 100 % cuesta $3,75 USD el valor total promedio por mes que la Empresa ABC est´a pagando debido a ineﬁciencias es $12.340,00 USD. Es decir, al a˜no se estima que paga $150.000,00 USD, por lo que se observa una gran oportunidad de mejora. Cabe indicar que ´en ´mano

este valor toma cuenta unicamente la de obra directa, no se ha incluido los otros valores indicados anteriormente, por lo que se concluye que la p´erdida es mucho mayor pero imposible de calcular con la informaci´on que posee actualmente la Empresa ABC.

**2.3. Notaci´atica del Problema**

**on Matem´**

El problema a resolver en la Empresa ABC se describe y formula a continuaci´on:

Datos de entrada:

Se tiene un conjunto de 10 m´aquinas que representan c´elulas de producci´on de los procesos existentes en la l´ınea de inter´es.

Se tiene un conjunto de art´ıculos (tareas) con cantidades determinadas a producir en esa l´ınea en el d´ıa de inter´es. Cada art´ıculo tiene rutas establecidas con operaciones determinadas que dependen del producto a fabricar.

Objetivo:

***ICM 2-48 ESPOL***

Minimizar el tiempo total de proceso de una secuencia factible. Donde el tiempo total del proceso es el m´aximo tiempo necesario para fabricar todos los art´ıculos.

*Mincma*x = *MaxNj*=1(*Cj*). En donde *C*j es el tiempo total requerido para completar la tarea j y se calcula Cj = *Snj,*j + *dnj,*j

Restricciones:

*Oh*j − *Oi*j ≥ *di*j para todo (*i*, *j*)(*h.j*)*εA*

→

*Cma*x − *Oi*j ≥ *di*j para todo (*i*, *j*)*ε*N

*Oi*j − *Oi*k ≥ *di*k ´o *Oi*k − *Oi*j ≥ *di*j para todo (*i*, *k*)*y*(*i*, *j*)*,*i =1*,...,*m

*Oi*j ≥ 0 para todo (*i*, *j*)*ε*N

La primera restricci´on asegura que la operaci´on *(h,j*) no puede iniciar antes que se complete la operaci´on *(i,j)*. La tercera restricci´on se conoce como la restricci´on disjunta, asegura que se respete el orden de diferentes tareas que se procesan en una misma m´aquina.

**2.4. Estructura del algoritmo de soluci ´on propuesto**

El algoritmo ha sido desarrollado en C++ usando como base el framework de algoritmos gen´eticos GALIB desarrollado por Matthew Wall en el Instituto Tecnol´ogico de Massachussets (MIT) [1].

**2.4.1. Codiﬁcaci ´on de los Genes**

Para el caso espec´ıﬁco del problema de job shop, existen dos diferentes t´ecnicas para representar el problema; la primera es una representaci´on indirecta la cual codiﬁca las instrucciones a trav´es de un constructor de secuencia; por ejemplo, la permutaci´on del orden de trabajos y la priorizaci´on de las reglas de secuenciamiento, en estos casos el constructor de secuencia asegura la validez del programa producido. La segunda t´ecnica es la representaci´on directa la cual codiﬁca la secuencia propiamente dicha, como ejemplo de ´esta t´ecnica se puede citar la codiﬁcaci´on de los

***ICM 2-49 ESPOL***

tiempos de ﬁnalizaci´on de cada operaci´on o los tiempos de inicio de cada operaci´on, en ´esta t´ecnica no toda la codiﬁcaci´on representa una secuencia v´alida.

La codiﬁcaci´on usada para resolver el problema planteado en el presente proyecto de titulaci´on es la *permutaci´o*n *ordenad*a *d*e *trabajos*. A continuaci´on se explica ´esta representaci´on mediante un ejemplo did´actico:

En un problema de 3 tareas con 4 m´aquinas se tiene:

Tareas = *{*J1 , J2 , J3}

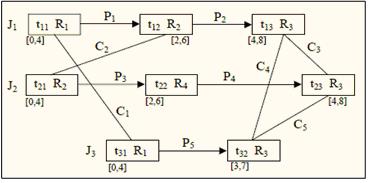
J1 = *{*t11(R1), t12(R2), t13(R3)}J2 = *{*t21(R2), t22(R4), t23(R3)}J3 = *{*t31(R1), t32(R3)}

Recursos = *{*R1, R2, R3, R4}

Du1j = Du2j = 2, Du3j = 3

Tmin = 0, Tmax = 10

En la Figura 2.2 se muestra un grafo disjunto que representa gr´aﬁcamente el problema de ejemplo.



En la Tabla 2.4 se presenta una soluci´on factible al problema.

Tabla 2.4: *Un*a *soluci´o*n *a*l *problem*a *plantead*o

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Recursos | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| R1 | *t*31 | |  | *t*1 | 1 |  |  |  |  |  |  |
| R2 | *t*21 | |  |  |  | *t*1 | 2 |  |  |  |  |
| R3 |  |  |  | *t*32 | |  | *t*2 | 3 | *t*1 | 3 |  |
| R4 |  |  | *t*22 | |  |  |  |  |  |  |  |

La representaci´on para un algoritmo gen´etico ser´ıa:  
Permutaci´on *Ordenad*a de las operaciones:

***ICM 2-50 ESPOL on Log´ıstica***

*{*t21, t31, t22, t11, t32, t12, t13, t23}

Reducci´on al n´umero de tareas:

*{*2, 3, 2, 1, 3, 1, 1, 2}

**2.4.2. N ´umero de Generaciones**

Las generaciones permiten al algoritmo gen´etico evolucionar obteniendo cada vez mejores cromosomas, por lo tanto es importante ﬁjar un n´umero adecuado de generaciones para conseguir un buen resultado. De la bibliograf´ıa revisada se obtiene que un n´umero adecuado para ´este tama˜no de problema es 5000, de ´esta forma nos aseguramos que el problema converja a la soluci´on ´optima.

**2.4.3. Tama ˜on**

**no de la Poblaci ´**

El tama˜no de la poblaci´on deﬁne el n´umero de cromosomas que existir´an en cada generaci´on. Si no existen suﬁcientes cromosomas, el algoritmo no evoluciona. De igual manera, si existen muchos cromosomas, se llega a un punto en el que los resultados no mejoran. Este tama˜no es directamente proporcional al n´umero de operaciones. Debido a que en el problema de la Empresa ABC el n´umero de operaciones es 10 se us´o la relaci´on tama˜no de la poblaci´on es igual al cuadrado del n´umero de operaciones, por lo tanto el tama˜no de la poblaci´on ser´a 100.

**2.4.4. Operador de Mutaci ´on**

Para el algoritmo propuesto en el presente proyecto de titulaci´on se usan dos operadores de mutaci´on.

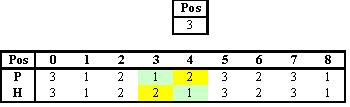
Operador de mutaci´on simple.

Operador de mutaci´on combinada.

La mutaci´on simple consiste en seleccionar aleatoriamente un n´umero entre 0 (cero) y el n´umero total de genes del cromosoma el cual identiﬁca la posici´on de mutaci´on. En la Figura 2.3 se intercambia los valores de los genes de la posici´on seleccionada con la posici´on inmediatamente siguiente.

La mutaci´on combinada consiste en:

***ICM 2-51 ESPOL***



Seleccionar al azar un n´umero y veriﬁcar si el n´umero es menor a la probabilidad de mutaci´on ( %). Si es menor, continuar con el siguiente paso, caso contrario no realizar la mutaci´on.

Seleccionar al azar un n´umero. Si es menor al 50 %, realizar la mutaci´on simple, para ello se seleccionan al azar dos posiciones del cromosoma y se intercambian sus valores, sino realizar una mutaci´on de desplazamiento con inversi´on de genes o sin ella, para esto se generan otros n´umeros al azar. Para el presente estudio el algoritmo de ´este m´etodo de mutaci´on se presenta a continuaci´on:

if nrandom ≤ *probMutacio*n then  
if *nrando*m ≤ 50 % then  
*intercambio*(*posicionrandom*1, *posicionrandom*2)

else

*nodos Random*

←

*SeleccionanNodos*(*posicionrandom*)

*inverti*r *Random*(≤ 50 %, 0, 1)

←

*InsertarnNodos*(*posicionrandom*, *invertir*)

end if  
end if

**2.4.5. Probabilidad de Mutaci ´on**

La probabilidad de mutaci´on es un par´ametro dado en el algoritmo que se encon

trar´a mediante una prueba experimental. Para ﬁjar el rango inicial de la probabilidad

de mutaci´on se tendr´a en cuenta el siguiente criterio:

La probabilidad de mutaci´on trata de impedir que la b´usqueda del algoritmo gen´etico caiga en optimos locales por eso es conveniente que ocurra ocasion-´

***ICM 2-52 ESPOL***

almente. No es bueno sin embargo, que la mutaci´on ocurra continuamente, ya que la b´usqueda del gen´etico pasa de ser *inteligent*e a b´usqueda aleatoria. Para el presente estudio se comenzar´a con un valor de 1/L, donde L es la longitud del cromosoma.

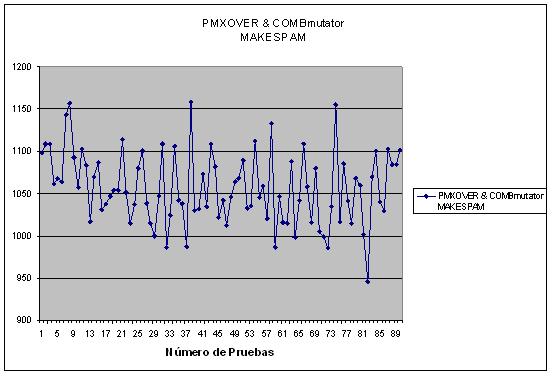
Se procede a calibrar la probabilidad de mutaci´on para el algoritmo propuesto, es decir, encontrar el valor de dicho par´ametro con el cual se obtiene un menor makespan. Para el efecto se realizan 90 experimentos con el problema est´andar FT10, donde se mantiene constante la probabilidad de reemplazo, la poblaci´on, el n´umero de generaciones, la probabilidad de cruce y la semilla para la generaci´on de los n´umeros aleatorios. Los resultados se indican en la Tabla 2.5.

Tabla 2.5: *Tabl*a *d*e *calibraci´o*n *d*e *probabilida*d *d*e *mutaci´o*n

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prueba | Reemplz | Poblaci´on | Generaci´on | Mutaci´on | Cruce | Seed | PMXOVER & |
|  |  |  |  |  |  |  | COMBmutator |
|  |  |  |  |  |  |  | MAKESPAM |
| 1 | 0,86 | 100 | 5000 | 0,01 | 0,60 | 99 | 1098 |
| 2 | 0,86 | 100 | 5000 | 0,011 | 0,60 | 99 | 1109 |
| 3 | 0,86 | 100 | 5000 | 0,012 | 0,60 | 99 | 1109 |
| 4 | 0,86 | 100 | 5000 | 0,013 | 0,60 | 99 | 1061 |
| 5 | 0,86 | 100 | 5000 | 0,014 | 0,60 | 99 | 1068 |
| 6 | 0,86 | 100 | 5000 | 0,015 | 0,60 | 99 | 1064 |
| 7 | 0,86 | 100 | 5000 | 0,016 | 0,60 | 99 | 1143 |
| 8 | 0,86 | 100 | 5000 | 0,017 | 0,60 | 99 | 1157 |
| 9 | 0,86 | 100 | 5000 | 0,018 | 0,60 | 99 | 1093 |
| 10 | 0,86 | 100 | 5000 | 0,019 | 0,60 | 99 | 1057 |
| 11 | 0,86 | 100 | 5000 | 0,02 | 0,60 | 99 | 1103 |
| 12 | 0,86 | 100 | 5000 | 0,021 | 0,60 | 99 | 1083 |
| 13 | 0,86 | 100 | 5000 | 0,022 | 0,60 | 99 | 1017 |
| 14 | 0,86 | 100 | 5000 | 0,023 | 0,60 | 99 | 1070 |
| 15 | 0,86 | 100 | 5000 | 0,024 | 0,60 | 99 | 1087 |
| 16 | 0,86 | 100 | 5000 | 0,025 | 0,60 | 99 | 1031 |
| 17 | 0,86 | 100 | 5000 | 0,026 | 0,60 | 99 | 1038 |
| 18 | 0,86 | 100 | 5000 | 0,027 | 0,60 | 99 | 1047 |
| 19 | 0,86 | 100 | 5000 | 0,028 | 0,60 | 99 | 1054 |
| 20 | 0,86 | 100 | 5000 | 0,029 | 0,60 | 99 | 1054 |
| 21 | 0,86 | 100 | 5000 | 0,03 | 0,60 | 99 | 1114 |
| 22 | 0,86 | 100 | 5000 | 0,031 | 0,60 | 99 | 1051 |
| 23 | 0,86 | 100 | 5000 | 0,032 | 0,60 | 99 | 1015 |
| 24 | 0,86 | 100 | 5000 | 0,033 | 0,60 | 99 | 1037 |
| 25 | 0,86 | 100 | 5000 | 0,034 | 0,60 | 99 | 1080 |
| 26 | 0,86 | 100 | 5000 | 0,035 | 0,60 | 99 | 1100 |
| 27 | 0,86 | 100 | 5000 | 0,036 | 0,60 | 99 | 1039 |
| 28 | 0,86 | 100 | 5000 | 0,037 | 0,60 | 99 | 1015 |
| 29 | 0,86 | 100 | 5000 | 0,038 | 0,60 | 99 | 1000 |
| 30 | 0,86 | 100 | 5000 | 0,039 | 0,60 | 99 | 1047 |
| 31 | 0,86 | 100 | 5000 | 0,04 | 0,60 | 99 | 1109 |
| 32 | 0,86 | 100 | 5000 | 0,041 | 0,60 | 99 | 986 |
| 33 | 0,86 | 100 | 5000 | 0,042 | 0,60 | 99 | 1024 |
| 34 | 0,86 | 100 | 5000 | 0,043 | 0,60 | 99 | 1106 |
| 35 | 0,86 | 100 | 5000 | 0,044 | 0,60 | 99 | 1042 |
| 36 | 0,86 | 100 | 5000 | 0,045 | 0,60 | 99 | 1038 |
| 37 | 0,86 | 100 | 5000 | 0,046 | 0,60 | 99 | 987 |
| 38 | 0,86 | 100 | 5000 | 0,047 | 0,60 | 99 | 1159 |
| 39 | 0,86 | 100 | 5000 | 0,048 | 0,60 | 99 | 1030 |
| 40 | 0,86 | 100 | 5000 | 0,049 | 0,60 | 99 | 1032 |
| 41 | 0,86 | 100 | 5000 | 0,05 | 0,60 | 99 | 1073 |
| 42 | 0,86 | 100 | 5000 | 0,051 | 0,60 | 99 | 1034 |
| 43 | 0,86 | 100 | 5000 | 0,052 | 0,60 | 99 | 1109 |
| 44 | 0,86 | 100 | 5000 | 0,053 | 0,60 | 99 | 1082 |
| 45 | 0,86 | 100 | 5000 | 0,054 | 0,60 | 99 | 1022 |
| 46 | 0,86 | 100 | 5000 | 0,055 | 0,60 | 99 | 1042 |
| 47 | 0,86 | 100 | 5000 | 0,056 | 0,60 | 99 | 1012 |
| 48 | 0,86 | 100 | 5000 | 0,057 | 0,60 | 99 | 1046 |
| 49 | 0,86 | 100 | 5000 | 0,058 | 0,60 | 99 | 1064 |
| 50 | 0,86 | 100 | 5000 | 0,059 | 0,60 | 99 | 1068 |
| 51 | 0,86 | 100 | 5000 | 0,06 | 0,60 | 99 | 1089 |
| 52 | 0,86 | 100 | 5000 | 0,061 | 0,60 | 99 | 1033 |
| 53 | 0,86 | 100 | 5000 | 0,062 | 0,60 | 99 | 1035 |
| 54 | 0,86 | 100 | 5000 | 0,063 | 0,60 | 99 | 1112 |
| 55 | 0,86 | 100 | 5000 | 0,064 | 0,60 | 99 | 1045 |
| 56 | 0,86 | 100 | 5000 | 0,065 | 0,60 | 99 | 1059 |
| 57 | 0,86 | 100 | 5000 | 0,066 | 0,60 | 99 | 1020 |
| 58 | 0,86 | 100 | 5000 | 0,067 | 0,60 | 99 | 1133 |
| 59 | 0,86 | 100 | 5000 | 0,068 | 0,60 | 99 | 986 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 60 | 0,86 | 100 | 5000 | 0,069 | 0,60 | 99 | 1046 |
| 61 | 0,86 | 100 | 5000 | 0,07 | 0,60 | 99 | 1016 |
| 62 | 0,86 | 100 | 5000 | 0,071 | 0,60 | 99 | 1015 |
| 63 | 0,86 | 100 | 5000 | 0,072 | 0,60 | 99 | 1088 |
| 64 | 0,86 | 100 | 5000 | 0,073 | 0,60 | 99 | 998 |
| 65 | 0,86 | 100 | 5000 | 0,074 | 0,60 | 99 | 1042 |
| 66 | 0,86 | 100 | 5000 | 0,075 | 0,60 | 99 | 1109 |
| 67 | 0,86 | 100 | 5000 | 0,076 | 0,60 | 99 | 1058 |
| 68 | 0,86 | 100 | 5000 | 0,077 | 0,60 | 99 | 1016 |
| 69 | 0,86 | 100 | 5000 | 0,078 | 0,60 | 99 | 1080 |
| 70 | 0,86 | 100 | 5000 | 0,079 | 0,60 | 99 | 1005 |
| 71 | 0,86 | 100 | 5000 | 0,08 | 0,60 | 99 | 999 |
| 72 | 0,86 | 100 | 5000 | 0,081 | 0,60 | 99 | 985 |
| 73 | 0,86 | 100 | 5000 | 0,082 | 0,60 | 99 | 1034 |
| 74 | 0,86 | 100 | 5000 | 0,083 | 0,60 | 99 | 1155 |
| 75 | 0,86 | 100 | 5000 | 0,084 | 0,60 | 99 | 1017 |
| 76 | 0,86 | 100 | 5000 | 0,085 | 0,60 | 99 | 1085 |
| 77 | 0,86 | 100 | 5000 | 0,086 | 0,60 | 99 | 1041 |
| 78 | 0,86 | 100 | 5000 | 0,087 | 0,60 | 99 | 1015 |
| 79 | 0,86 | 100 | 5000 | 0,088 | 0,60 | 99 | 1068 |
| 80 | 0,86 | 100 | 5000 | 0,089 | 0,60 | 99 | 1060 |
| 81 | 0,86 | 100 | 5000 | 0,09 | 0,60 | 99 | 1001 |
| 82 | 0,86 | 100 | 5000 | 0,091 | 0,60 | 99 | 946 |
| 83 | 0,86 | 100 | 5000 | 0,092 | 0,60 | 99 | 1070 |
| 84 | 0,86 | 100 | 5000 | 0,093 | 0,60 | 99 | 1100 |
| 85 | 0,86 | 100 | 5000 | 0,094 | 0,60 | 99 | 1040 |
| 86 | 0,86 | 100 | 5000 | 0,095 | 0,60 | 99 | 1029 |
| 87 | 0,86 | 100 | 5000 | 0,096 | 0,60 | 99 | 1103 |
| 88 | 0,86 | 100 | 5000 | 0,097 | 0,60 | 99 | 1084 |
| 89 | 0,86 | 100 | 5000 | 0,098 | 0,60 | 99 | 1084 |
| 90 | 0,86 | 100 | 5000 | 0,099 | 0,60 | 99 | 1102 |

En la Figura 2.4 se presenta un gr´aﬁco del desempe˜no de las pruebas en funci´on del valor de makespan.



El mejor valor de makespan manteniendo inalterables los otros par´ametros es de 946 para una probabilidad de mutaci´on de 0,091 por lo tanto se trabajar´a con ´esta proababilidad para el resto de calibraciones.

***ICM 2-54 ESPOL***

**2.4.6. Operador de Cruce**

El algoritmo desarrollado tiene la posibilidad de usar dos operadores de cruce: el PMX y JOX. En las pruebas de los problemas est´andar que se indican en el cap´ıtulo 3 se usar´an los dos operadores para conocer el que presente mejor desempe˜no y aplicarlo en la soluci´on de la Empresa ABC.

**2.4.7. Probabilidad de Cruce**

La probabilidad de cruce es un par´ametro dado en el algoritmo que se encontrar´a mediante una prueba experimental. Para ﬁjar el rango inicial de la probabilidad de cruce se tendr´a en cuenta el siguiente criterio:

La probabilidad de cruce debe estar entre el 60 % al 95 %, de ´esta forma se asegura que el problema no llegue a convergencia de forma lenta y algunos individuos pasen a la siguiente generaci´on sin cruzarse.

Para encontrar el valor de ´este par´ametro se usa el problema est´andar FT10 y se usa el operador PMX. Se realizaron 36 pruebas variando la probabilidad de cruce en 1 % desde el 60 % al 95 % obteniendo para cada una el makespan.

El mejor valor de ´este par´ametros es 0,6 con un makespan de 946 (2 % de error) como se detalla en la Tabla 2.6:

Tabla 2.6: *Tabl*a *d*e *calibraci´o*n *d*e *probabilida*d *d*e *cruc*e

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prueba | Reemplz | Poblaci´on | Generaci´on | Mutaci´on | Cruce | Seed | PMXOVER & |
|  |  |  |  |  |  |  | COMBmutator |
|  |  |  |  |  |  |  | MAKESPAM |
| 1 | 0,86 | 100 | 5000 | 0,091 | 0,95 | 99 | 1024 |
| 2 | 0,86 | 100 | 5000 | 0,091 | 0,94 | 99 | 1020 |
| 3 | 0,86 | 100 | 5000 | 0,091 | 0,93 | 99 | 981 |
| 4 | 0,86 | 100 | 5000 | 0,091 | 0,92 | 99 | 1013 |
| 5 | 0,86 | 100 | 5000 | 0,091 | 0,91 | 99 | 1039 |
| 6 | 0,86 | 100 | 5000 | 0,091 | 0,90 | 99 | 1045 |
| 7 | 0,86 | 100 | 5000 | 0,091 | 0,89 | 99 | 1043 |
| 8 | 0,86 | 100 | 5000 | 0,091 | 0,88 | 99 | 1029 |
| 9 | 0,86 | 100 | 5000 | 0,091 | 0,87 | 99 | 1046 |
| 10 | 0,86 | 100 | 5000 | 0,091 | 0,86 | 99 | 1032 |
| 11 | 0,86 | 100 | 5000 | 0,091 | 0,85 | 99 | 1033 |
| 12 | 0,86 | 100 | 5000 | 0,091 | 0,84 | 99 | 1011 |
| 13 | 0,86 | 100 | 5000 | 0,091 | 0,83 | 99 | 1018 |
| 14 | 0,86 | 100 | 5000 | 0,091 | 0,82 | 99 | 1067 |
| 15 | 0,86 | 100 | 5000 | 0,091 | 0,81 | 99 | 991 |
| 16 | 0,86 | 100 | 5000 | 0,091 | 0,80 | 99 | 991 |
| 17 | 0,86 | 100 | 5000 | 0,091 | 0,79 | 99 | 1055 |
| 18 | 0,86 | 100 | 5000 | 0,091 | 0,78 | 99 | 1063 |
| 19 | 0,86 | 100 | 5000 | 0,091 | 0,77 | 99 | 997 |
| 20 | 0,86 | 100 | 5000 | 0,091 | 0,76 | 99 | 1014 |
| 21 | 0,86 | 100 | 5000 | 0,091 | 0,75 | 99 | 1035 |
| 22 | 0,86 | 100 | 5000 | 0,091 | 0,74 | 99 | 1038 |
| 23 | 0,86 | 100 | 5000 | 0,091 | 0,73 | 99 | 1076 |
| 24 | 0,86 | 100 | 5000 | 0,091 | 0,72 | 99 | 1035 |
| 25 | 0,86 | 100 | 5000 | 0,091 | 0,71 | 99 | 1039 |
| 26 | 0,86 | 100 | 5000 | 0,091 | 0,70 | 99 | 1061 |
| 27 | 0,86 | 100 | 5000 | 0,091 | 0,69 | 99 | 1017 |
| 28 | 0,86 | 100 | 5000 | 0,091 | 0,68 | 99 | 970 |
| 29 | 0,86 | 100 | 5000 | 0,091 | 0,67 | 99 | 1060 |
| 30 | 0,86 | 100 | 5000 | 0,091 | 0,66 | 99 | 1012 |
| 31 | 0,86 | 100 | 5000 | 0,091 | 0,65 | 99 | 1082 |

32 0,86

100

5000

0,091

0,64

99

1013

33 0,86

100

5000

0,091

0,63

99

1015

34 0,86

100

5000

0,091

0,62

99

993

35 0,86

100

5000

0,091

0,61

99

1056

36 0,86

100

5000

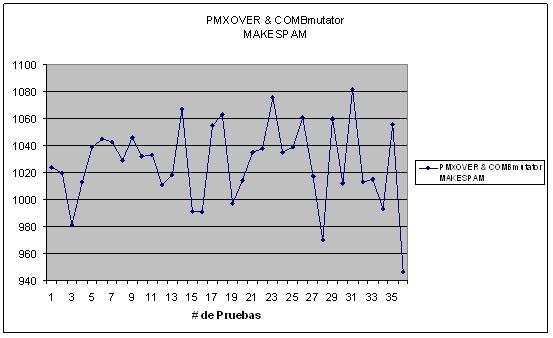
0,091

0,60

99

946

En la Figura 2.5 se presenta un gr´aﬁco del desempe˜no de las pruebas en funci´on del valor de makespan.



El mejor valor de makespam manteniendo inalterables los otros par´ametros es de 946 para una probabilidad de cruce de 0,6 por lo tanto se trabajar´a con ese porcentaje para el resto de calibraciones.

**2.4.8. Probabilidad de Reemplazo**

La probabilidad de reemplazo es necesaria ya que el algoritmo construido es del tipo STEADY-STATE, el cual utiliza el traslape de poblaciones. Para cada generaci´on solamente una porci´on de ´esta es reemplazada por sus hijos. Cuando la probabilidad de reemplazo es del 100 % solamente uno o dos genes son reemplazados, mientras que si la probabilidad de reemplazo es del 0 %, todos los individuos de la poblaci´on son reemplazados.Para la calibraci´on de la probabilidad de reemplazo se realizan 20 experimentos con el problema est´andar FT10, donde se mantiene

***ICM 2-56 ESPOL***

constante la poblaci´on, el n´umero de generaciones, la probabilidad de mutaci´on, la probabilidad de cruce y la semilla para la generaci´on de los n´umeros aleatorios. Los resultados se indican en la Tabla 2.7.

Tabla 2.7: *Tabl*a *d*e *calibraci´o*n *d*e *probabilida*d *d*e *reemplaz*o

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prueba | Reemplz | Poblaci´on | Generaci´on | Mutaci´on | Cruce | Seed | PMXOVER & |
|  |  |  |  |  |  |  | COMBmutator |
|  |  |  |  |  |  |  | MAKESPAM |
| 1 | 0,8 | 100 | 5000 | 0,091 | 0,60 | 99 | 1034 |
| 2 | 0,81 | 100 | 5000 | 0,091 | 0,60 | 99 | 1068 |
| 3 | 0,82 | 100 | 5000 | 0,091 | 0,60 | 99 | 1031 |
| 4 | 0,83 | 100 | 5000 | 0,091 | 0,60 | 99 | 1108 |
| 5 | 0,84 | 100 | 5000 | 0,091 | 0,60 | 99 | 1076 |
| 6 | 0,85 | 100 | 5000 | 0,091 | 0,60 | 99 | 1112 |
| 7 | 0,86 | 100 | 5000 | 0,091 | 0,60 | 99 | 946 |
| 8 | 0,87 | 100 | 5000 | 0,091 | 0,60 | 99 | 1048 |
| 9 | 0,88 | 100 | 5000 | 0,091 | 0,60 | 99 | 997 |
| 10 | 0,89 | 100 | 5000 | 0,091 | 0,60 | 99 | 982 |
| 11 | 0,9 | 100 | 5000 | 0,091 | 0,60 | 99 | 1040 |
| 12 | 0,91 | 100 | 5000 | 0,091 | 0,60 | 99 | 1028 |
| 13 | 0,92 | 100 | 5000 | 0,091 | 0,60 | 99 | 1123 |
| 14 | 0,93 | 100 | 5000 | 0,091 | 0,60 | 99 | 1080 |
| 15 | 0,94 | 100 | 5000 | 0,091 | 0,60 | 99 | 1095 |
| 16 | 0,95 | 100 | 5000 | 0,091 | 0,60 | 99 | 1042 |
| 17 | 0,96 | 100 | 5000 | 0,091 | 0,60 | 99 | 1024 |
| 18 | 0,97 | 100 | 5000 | 0,091 | 0,60 | 99 | 1049 |
| 19 | 0,98 | 100 | 5000 | 0,091 | 0,60 | 99 | 1116 |
| 20 | 0,99 | 100 | 5000 | 0,091 | 0,60 | 99 | 1056 |

El mejor makespan vuelve a ser 946 con un porcentaje de reemplazo de 0,86 por tanto el algoritmo est´a calibrado y listo para ser usado en los problemas est´andar y en el problema de inter´es.

**2.4.9. Archivo de Par´ametros**

El archivo de par´ametros contiene la informaci´on con la que va a operar el algoritmo. El nombre del archivo debe ser siempre *entrad*a *jss.tx*t y los datos que contiene son:

Probabilidad de cruce  
Probabilidad de mutaci´on  
Tama˜no de la poblaci´on  
N´umero de generaciones  
Probabilidad de reemplazo  
Semilla de n´umeros aleatorios

**2.4.10. Archivo de Resultados**

El archivo de resultados contiene la informaci´on de la soluci´on del problema, el nombre del archivo siempre es *resultado*s *jss.txt*; contiene la siguiente informaci´on:

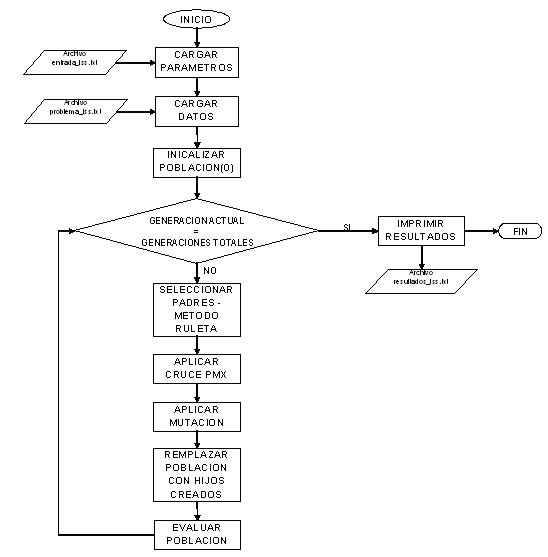
MakeSpan M´ınimo

Secuencia Planiﬁcada

Estad´ıstica de toda la ejecuci´on del algoritmo, entre otros valores tenemos, n´umero de cruces, n´umero de mutaciones, promedio de todos los makespan, etc.

**2.4.11. Diagrama de Flujo**

En la Figura 2.6 se presenta el diagrama de ﬂujo del algoritmo propuesto para resolver el problema de inter´es.



**CAPITULO 3**

**PRUEBAS Y RESULTADOS CON PROBLEMAS ESTANDAR**

A continuaci´on, se presenta los resultados obtenidos luego de aplicar el algoritmo gen´etico construido a los problemas est´andar FT20, FT10, LA24, LA27 y FT6.

**3.1. Problemas Est´**

**andar y Hardware usado**

Para la evaluaci´on del algoritmo desarrollado durante el presente proyecto de titulaci´on se ha seleccionado 5 problemas est´andar de estudio de la comunidad cient´ıﬁca sobre el JOB SHOP SCHEDULING PROBLEM disponibles en Internet en la librer´ıa OR [2]. Los experimentos han sido ejecutados en una LAPTOP Core 2 DUO T5500, 1.66Ghz, con una memoria RAM de 2.0 GB, bajo Windows XP Service Pack 3. Se realiz´o 400 ejecuciones para cada instancia (problema est´andar). En cada problema est´andar con los par´ametros calibrados en el cap´ıtulo 2 se evalu´o las 4 combinaciones posibles de operador de mutaci´on y operador de cruce:

Operador de cruce JOX y mutaci´on simple

Operador de cruce JOX y mutaci´on combinada

Operador de cruce PMX y mutaci´on simple

Operador de cruce PMX y mutaci´on combinada

***ICM 3-59 ESPOL***

Para cada una de las 4 combinaciones se realizan 100 pruebas y se guarda la mejor soluci´on encontrada. En la Tabla 3.1 se presentan los resultados obtenidos por el algoritmo en las instancias evaluadas y en el anexo 1 se pueden encontrar los datos completos.

Tabla 3.1: *Mejore*s *resultado*s *po*r *instanci*a

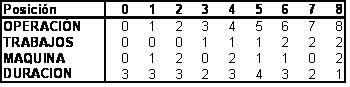
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instancia | Dimensi´on | Optimo Conocido | jss | Error |
| FT20 | 20 × 5 | 1165 | 1183 | 2 % |
| FT10 | 10 × 10 | 930 | 946 | 2 % |
| LA24 | 15 × 10 | 935 | 984 | 5 % |
| LA27 | 20 × 10 | 1235 | 1323 | 7 % |
| FT6 | 6 × 6 | 54 | 55 | 2 % |

El algoritmo construido tiene un error promedio del 3 %.

**3.2. Ejemplo de decodiﬁcaci ´**

**on y funcionamiento del algoritmo**

Para entender el funcionamiento del algoritmo propuesto y con ﬁnes did´acticos se realizaron pruebas con un problema est´andar de 3 x 3. Para ´este problema se desarroll´o la respectiva codiﬁcaci´on de la soluci´on que se presenta en la Figura 3.1.

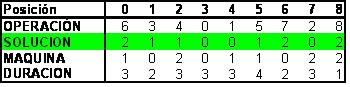


Dada una soluci´on factible obtenida durante la ejecuci´on del algoritmo, el pro-grama busca el orden de operaciones, las m´aquinas y la duraci´on de cada operaci´on para continuar con la ejecuci´on del algoritmo. En la Figura 3.2 se presenta una soluci´on:

La codiﬁcaci´on del problema seg´un lo explicado es:

Como se puede observar la secuencia planiﬁcada empieza con 2 lo que signiﬁca que se debe procesar la primera operaci´on del trabajo n´umero 2. Seg´un la codiﬁcaci´on propuesta ´esta es la operaci´on 6 que se realiza en la m´aquina 1 y tiene un tiempo de duraci´on de 3. De igual forma se procede para el resto de la secuencia. Esta

***ICM 3-60 ESPOL***



informaci´on es almacenada en arreglos auxiliares que se usar´an para el c´alculo del MakeSpan. Los arreglos son:

OP Arreglo que almacena la secuencia de las operaciones

MQ Arreglo que almacena la secuencia de m´aquinas

TR Arreglo que almacena la secuencia de trabajos

TI Arreglo que almacena los tiempos de operaci´on

El c´alculo del MakeSpan en el algoritmo desarrollado se sustenta en los tiempos de inicio y de ﬁn de cada una de las operaciones. Para ello es necesario usar dos arreglos auxiliares que contendr´an ´esta informaci´on, TINICIO y TFIN. Primero inicializamos los arreglos de tiempo de inicio y ﬁn de las operaciones

while i < *NumeroOperacione*s do  
*TINICIO*[*i*]=0  
*TFIN*[*i*]= *−*1

end while

El algoritmo de c´alculo es el siguiente: for i = 0 to NumeroOperaciones do Si es la primera operaci´on el tiempo de ﬁn es el tiempo de ejecuci´on de la operaci´on if i =0 then *TFIN*[*i*]= *TI*[*i*]

else

Auxiliar de M´aquinas  
*aux*1= *−*1  
for j = i − 1 to 0 do

if *MQ*[*i*]= *MQ*[*j*] then

***ICM 3-61 ESPOL on Log´ıstica***

*aux*1= *TFIN*[*j*]  
Salir del for

end if  
end for

Auxiliar de Trabajos  
*aux*2= *−*1  
for j = i − 1 to 0 do

if *TR*[*i*]= *TR*[*j*] then  
*aux*2= *TFIN*[*j*]  
Salir del for

end if  
end for  
if *aux*1= *aux*2 then

Es la primera operaci´on del trabajo y la m´aquina a´un no se ha utilizado

if *aux*1= *−*1 then  
*TINICIO*[*i*]=0  
*TFIN*[*i*]= *TI*[*i*]

else

*TINICIO*[*i*]= *aux*1  
*TFIN*[*i*]= *aux*1+ *TI*[*i*]

end if  
end if  
if aux1 > aux2 then

La m´aquina est´a disponible y la operaci´on inmediatamente anterior ya se ha terminado *TINICIO*[*i*]= *aux*1 *TFIN*[*i*]= *aux*1+ *TI*[*i*]

end if  
end if  
end for

El mayor valor del arreglo TFIN es el MakeSpan  
*mkspa*m =0

***ICM 3-62 ESPOL***

***on Log´ıstica***

while i < *NumeroOperacione*s do if *mkspa*m < *TFIN*[*i*] then *Mkspa*m = *TFIN*[*i*]

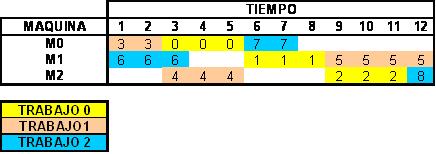
end if end while

El MakeSpan para el ejemplo es 12, tal como se resume en la Figura 3.3.



Figura 3.3: Makespam

Gr´aﬁcamente la secuenciaci´on se la representa en la Figura 3.4:



Los n´umeros dentro de los trabajos son las operaciones seg´un la secuencia planiﬁcada tal como se explic´o anteriormente.

**CAPITULO 4**

**APLICACION EN EL PROBLEMA DE ESTUDIO**

En ´este cap´ıtulo se presentan los resultados obtenidos luego de aplicar al problema objeto de la tesis, el algoritmo gen´etico construido.

**4.1. Descripci´on del Problema de Estudio**

Para la resoluci´on se tom´o la planiﬁcaci´on de un d´ıa donde exist´ıa la mayor cantidad de productos posibles y se procedi´o a resolver usando el algoritmo propuesto y desarrollado en el presente proyecto de titulaci´on.

En la Tabla 4.1, se presenta el cuadro donde se tiene el problema a resolver con los tiempos requeridos en cada c´elula de trabajo para poder fabricar las cantidades determinadas previamente en el plan maestro de producci´on. Cabe indicar que los tiempos requeridos se obtienen de las rutas existentes en el ERP.

Tabla 4.1: *Problem*a a *resolve*r *co*n *e*l *algoritm*o *propuest*o

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Producto | Pelado | Picado | Lavado | Procesar | LavadoEn-vases | PreparacLG | Envasado | Cerrado | Pasteurizado | Embalado |
| ACEITUNAVERDE PEPA 230 G DOY PACK |  |  | 9 | 35 |  | 1 | 17 | 4 |  | 2 |
| ACEITUNAVERDE RELLENA PIMIENTO 230 G DOY PACK |  |  | 9 | 35 |  | 1 | 17 | 4 |  | 2 |
| ALCACHOFAENTERA 430 G | 9 | 7 | 11 | 23 | 4 | 1 | 14 | 1 | 11 | 2 |
| ALCACHOFAMITADES 400 G. | 9 | 13 | 11 | 23 | 4 | 1 | 14 | 1 | 11 | 2 |
| ARVEJA 420 G. |  |  | 17 | 29 | 4 | 1 | 15 | 1 | 11 | 2 |
| CEBOLLITASEN VINAGRE200 G DOY PACK |  |  | 15 | 35 |  | 1 | 17 |  |  | 2 |
| CEBOLLITASEN VINAGRE250 G. |  |  | 16 | 36 | 6 | 1 | 14 | 1 | 11 | 2 |
| CHAMPI ˜NON EN VINAGRE4.1 KG |  |  | 7 | 15 | 3 | 1 | 31 | 3 |  | 1 |
| CHOCLITO 2.5 KG | 37 |  | 9 | 9 | 3 | 1 | 22 | 3 |  | 1 |
| CHOCLITO 240 G | 5 |  | 2 | 2 | 4 | 1 | 12 | 2 | 9 | 3 |
| CHOCLITO 810 G |  | 11 |  |  | 4 | 1 | 16 | 2 |  | 2 |
| ESPARRAGOBLANCO JUMBO FCO.350 | 7 |  | 3 | 19 | 4 | 1 | 14 | 2 | 8 | 2 |
| ESPARRAGOBLANCO 430 G. | 7 |  | 3 | 19 | 4 | 1 | 14 | 2 | 8 | 2 |
| ESPARRAGOBLANCO TROZO 430 G. |  | 3 | 3 | 19 | 4 | 1 | 14 | 2 | 8 | 2 |
| ESPARRAGOVERDE 430 G. | 7 |  | 3 | 19 | 4 | 1 | 14 | 2 | 8 | 2 |
| FREJOL NEGRO 425 GR. |  |  | 19 | 17 | 4 | 1 | 12 | 1 | 11 | 2 |
| FREJOL ROJO 425 GR. |  |  | 15 | 17 | 4 | 1 | 12 | 1 | 11 | 2 |
| MENESTRA DELENTEJA 425 G |  |  | 19 | 18 | 4 | 1 | 12 | 1 | 11 | 2 |
| PALMITO 410 GEO NACIONAL | 5 | 2 | 4 | 12 | 4 | 1 | 9 | 1 | 12 | 2 |
| PALMITO 440 GFCO NACIONAL | 6 | 2 | 4 | 13 | 4 | 1 | 10 | 1 | 12 | 2 |
| PALMITO 810 GEO NACIONAL | 9 | 3 | 5 | 15 | 4 | 1 | 12 | 1 | 12 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PALMITO CULTIVADO ”B”410 G. | 5 | 2 | 4 | 12 | 4 | 1 | 9 | 1 | 12 | 2 |
| PALMITO CULTIVADO C¸ ”810 G. | 9 | 3 | 5 | 15 | 4 | 1 | 12 | 1 | 12 | 1 |
| PALMITO ENMITADES 410 G |  | 9 | 4 | 9 | 4 | 1 | 9 | 1 | 12 | 2 |
| PALMITOMEDALLONES810 G |  | 9 | 5 | 11 | 4 | 1 | 12 | 1 | 12 | 1 |
| PALMITO MEDIA LUNA 410 G. |  | 8 | 4 | 9 | 4 | 1 | 9 | 1 | 12 | 2 |
| PALMITO TROZOS 410 GR. |  | 11 | 4 | 9 | 4 | 1 | 9 | 1 | 12 | 2 |
| PEPINILLOAGRIDULCE220 G DOY PACK |  | 4 | 2 | 3 |  | 1 | 12 | 4 |  | 3 |
| PEPINILLOAGRIDULCE 4.1KG |  | 21 | 17 | 12 | 3 | 1 | 31 | 2 |  | 1 |
| PEPINILLOAGRIDULCE470 G. |  | 6 | 5 | 5 | 4 | 1 | 17 | 2 | 7 | 2 |
| PEPINILLO ENTERO 4.1 KG |  |  | 29 | 12 | 3 | 1 | 31 | 2 |  | 1 |
| PEPINILLOENTERO 220 G DOY PACK |  |  | 7 | 5 |  | 1 | 17 | 4 |  | 3 |
| PEPINILLO ENTERO 470 G |  |  | 14 | 3 | 4 | 1 | 12 | 2 | 7 | 2 |
| PI ˜NAS ENALMIBAR 600 G | 23 |  | 17 | 31 | 4 | 1 | 7 | 1 | 12 | 2 |

Para la codiﬁcaci´on del problema cada art´ıculo a fabricar es una tarea, que se numera secuencialmente desde el 0 (cero), con lo que se tiene la informaci´on de la Tabla 4.2.

Tabla 4.2: *Codiﬁcaci´o*n *d*e *la*s *Tarea*s

Producto

Tarea

ACEITUNA VERDE PEPA 230 G DOY PACK

0

ACEITUNA VERDE RELLENA PIMIENTO 230 G DOY PACK

1

ALCACHOFA ENTERA 430 G

2

ALCACHOFA MITADES 400 G.

3

ARVEJA 420 G.

4

CEBOLLITAS EN VINAGRE 200 G DOY PACK

5

CEBOLLITAS EN VINAGRE 250 G.

6

CHAMPI ˜

NON EN VINAGRE 4.1 KG

7

CHOCLITO 2.5 KG

8

CHOCLITO 240 G

9

CHOCLITO 810 G

10

ESPARRAGO BLANCO JUMBO FCO.350

11

ESPARRAGO BLANCO 430 G.

12

ESPARRAGO BLANCO TROZO 430 G.

13

ESPARRAGO VERDE 430 G.

14

FREJOL NEGRO 425 GR.

15

FREJOL ROJO 425 GR.

16

MENESTRA DE LENTEJA 425 G

17

PALMITO 410 G EO NACIONAL

18

PALMITO 440 G FCO NACIONAL

19

PALMITO 810 G EO NACIONAL

20

PALMITO CULTIVADO ”B”410 G.

21

PALMITO CULTIVADO C¸ ”810 G.

22

PALMITO EN MITADES 410 G

23

PALMITO MEDALLONES 810 G

24

PALMITO MEDIA LUNA 410 G.

25

PALMITO TROZOS 410 GR.

26

PEPINILLO AGRIDULCE 220 G DOY PACK

27

PEPINILLO AGRIDULCE 4.1 KG

28

PEPINILLO AGRIDULCE 470 G.

29

PEPINILLO ENTERO 4.1 KG

30

PEPINILLO ENTERO 220 G DOY PACK

31

PEPINILLO ENTERO 470 G

32

PI ˜

NAS EN ALMIBAR 600 G

33

**4.2. Resoluci´on del Problema de Estudio**

Se colocan los datos del problema y los par´ametros previamente calibrados en el archivo de texto que usa el algoritmo. Se realizan 26 pruebas, en cada una de las cuales se var´ıa la semilla para la generaci´on de los n´umeros aleatorios usados con los resultados presentados en la Tabla 4.3.

Tabla 4.3: *Resultado*s *d*e *l*a *soluci´o*n *a*l *problem*a

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PRUEBA | REEMPLAZO | POBLACION | GENERACIONES | MUTACION | CRUCE | SEED | MAKESPAM |
| 0 1 2 3 | 0,86 0,86 0,86 0,86 | 100 100 100 100 | 5000 5000 5000 5000 | 0,091 0,091 0,091 0,091 | 0,6 0,6 0,6 0,6 | 0 1 2 3 | 599 587 599 599 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 4 | 588 |
| 5 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 5 | 584 |
| 6 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 6 | 573 |
| 7 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 7 | 590 |
| 8 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 8 | 574 |
| 9 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 9 | 606 |
| 10 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 10 | 601 |
| 11 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 11 | 589 |
| 12 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 12 | 579 |
| 13 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 13 | 614 |
| 14 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 14 | 615 |
| 15 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 15 | 592 |
| 16 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 16 | 582 |
| 17 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 17 | 596 |
| 18 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 18 | 601 |
| 19 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 19 | 592 |
| 20 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 20 | 608 |
| 21 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 21 | 591 |
| 22 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 22 | 603 |
| 23 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 23 | 596 |
| 24 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 24 | 590 |
| 25 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 25 | 603 |
| 26 | 0,86 | 100 | 5000 | 0,091 | 0,6 | 26 | 572 |

El tiempo de soluci´on es de 2 minutos 47 segundos, obteni´endose el mejor makespan que result´o ser de 572. La secuencia de planiﬁcaci´on se indica en la Tabla 4.4. Esta secuencia de planiﬁcaci´on se entiende de la siguiente forma:

La primera tarea que se tiene que empezar a producir es la n´umero 13, de acuerdo a la Tabla 4.2 se observa que la primera tarea a ser realizada es ESPARRAGO BLANCO TROZO 430 G. En la Tabla 4.1 se observa que la primera operaci´on de ´esta tarea es el picado con una duraci´on de 3 minutos.

La siguiente tarea es 13 nuevamente, de la Tabla 4.2 se observa que la siguiente operaci´on para esa tarea es el lavado con una duraci´on de 3 minutos. De ´esta forma se analiza toda la respuesta obtenida.

En otras palabras, los n´umeros que se obtienen en el archivo de resultados equivalen a las tareas que deben ser procesadas. En la Tabla 4.1, se tiene que buscar la siguiente operaci´on de la tarea de inter´es. De ´esta forma, se codiﬁca la respuesta en un diagrama de Gantt que se indica a continuaci´on.

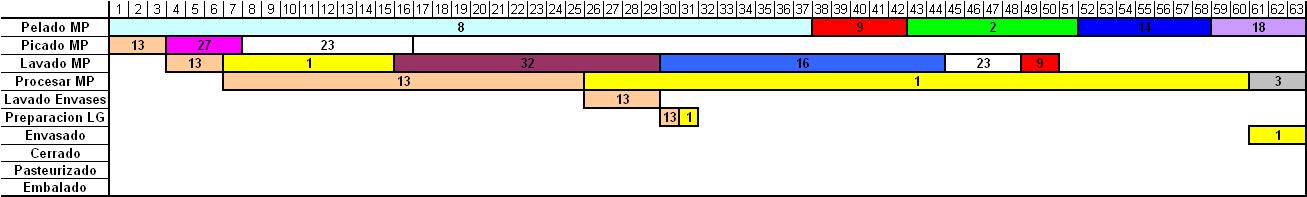


Tabla 4.4: *Secuenciaci´o*n *d*e *la*s *Tarea*s

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 13 | 8 | 13 | 27 | 9 | 13 | 2 | 1 | 1 | 32 | 16 | 23 | 14 | 18 | 23 | 18 | 13 | 1 | 9 |
| 10 | 22 | 32 | 21 | 23 | 18 | 18 | 24 | 18 | 28 | 32 | 25 | 1 | 13 | 8 | 3 | 1 | 21 | 11 | 0 |
| 21 | 23 | 18 | 22 | 20 | 17 | 23 | 21 | 19 | 21 | 16 | 18 | 8 | 20 | 16 | 2 | 21 | 4 | 20 | 19 |
| 25 | 1 | 21 | 7 | 13 | 9 | 32 | 12 | 16 | 24 | 23 | 27 | 29 | 8 | 15 | 10 | 28 | 19 | 8 | 16 |
| 10 | 9 | 13 | 7 | 19 | 20 | 7 | 25 | 30 | 8 | 26 | 25 | 10 | 9 | 26 | 24 | 15 | 32 | 32 | 12 |
| 20 | 25 | 27 | 7 | 15 | 24 | 11 | 14 | 28 | 22 | 7 | 8 | 2 | 14 | 14 | 9 | 28 | 12 | 22 | 20 |
| 2 | 26 | 33 | 18 | 31 | 15 | 28 | 14 | 23 | 12 | 10 | 20 | 28 | 20 | 15 | 22 | 12 | 14 | 23 | 27 |
| 7 | 12 | 19 | 14 | 27 | 22 | 6 | 3 | 26 | 11 | 4 | 19 | 26 | 24 | 3 | 21 | 31 | 15 | 31 | 2 |
| 17 | 24 | 12 | 2 | 2 | 14 | 16 | 8 | 11 | 21 | 30 | 24 | 4 | 7 | 30 | 15 | 5 | 22 | 20 | 31 |
| 31 | 10 | 11 | 4 | 2 | 33 | 27 | 33 | 11 | 16 | 11 | 30 | 0 | 25 | 12 | 29 | 29 | 18 | 11 | 9 |
| 30 | 3 | 19 | 29 | 0 | 22 | 26 | 4 | 26 | 30 | 31 | 9 | 27 | 9 | 19 | 4 | 11 | 25 | 24 | 29 |
| 28 | 32 | 17 | 26 | 17 | 3 | 4 | 2 | 17 | 29 | 26 | 16 | 4 | 6 | 3 | 15 | 32 | 32 | 6 | 3 |
| 29 | 6 | 25 | 5 | 17 | 13 | 25 | 0 | 6 | 23 | 22 | 33 | 3 | 29 | 17 | 29 | 0 | 6 | 30 | 2 |
| 5 | 33 | 19 | 12 | 3 | 33 | 28 | 6 | 0 | 21 | 20 | 33 | 17 | 5 | 17 | 13 | 18 | 19 | 24 | 6 |
| 3 | 33 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**4.3. Comparaci´on propuesta con el m´**

**on de la soluci ´etodo actual**

Para ver la efectividad del algoritmo propuesto se resuelve el mismo problema con el m´etodo actual, de lo que se obtienen los resultados de la Tabla 4.5.

Tabla 4.5: *Resultado*s *d*e *l*a *soluci´o*n *mediant*e *e*l *m´etod*o *actua*l

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Valor | Unidad | Observaci´on |
| Tiempo de soluci´on | 2 | d´ıa | Tiempo del jefe de planiﬁcaci´on, jefe de producci´on y gerente de producci´on |
| Makespam | 917 | minutos | Adicional faltan 60 minutos de comidas |

Como se puede observar el m´etodo desarrollado en el presente proyecto de titulaci´on presenta un makespam de 572 mientras que el desarrollado por el m´etodo tradicional de planiﬁcaci´on toma 917 minutos (es decir un 60 % adicional para ´este caso espec´ıﬁco). Este valor adicional representaba para la empresa un gasto en horas no laboradas pagadas al personal, ´este gasto innecesario disminuye debido al uso del programa desarrollado. Adicionalmente, el tiempo utilizado en la resoluci´on tambi´en es muy diferente por lo que se puede usar el nuevo m´etodo de planiﬁcaci´on ﬁnita de la producci´on para la empresa de inter´es. El tiempo requerido para obtener un

´

diagrama similar al de la Figura 4.1, es de aproximadamente 4 horas. Este es un tiempo demasiado largo, por lo que para presentar el programa de producci´on al personal operativo se lo har´a en un formato en el que se numera secuencialmente

***ICM 4-69 ESPOL***

las operaciones por tarea a producir. El formato toma alrededor de 15 minutos en prepararlo y es funcional para los operadores de la planta.

**4.4. Asignaci´on de la mano de obra al secuenciameinto**

El algoritmo desarrollado en el presente proyecto tambi´en sirve para la asignaci´on de la mano de obra directa necesaria para el proceso productivo. Con el archivo de la soluci´on se obtiene, en funci´on de las rutas almacenadas en el ERP, las personas necesarias en cada una de las operaciones de inter´es.

Cabe indicar la importancia de ´esta asignaci´on de forma autom´atica, ya que actualmente con el proceso de planiﬁcaci´on ﬁnita, el jefe de producci´on tiene que asignar el personal a las c´elulas de producci´on y crear los horarios de ingreso del personal manualmente, tarea que le toma mucho tiempo. Ahora, al tener el secuenciamiento en un archivo plano, se puede realizar la asignaci´on autom´aticamente.

**4.5. Otras oportundades de mejora**

Otro punto de inter´es es que como se observa en la Figura 4.1 existen celdas de producci´on que no est´an balanceadas. Las c´elulas que no est´an balanceadas son:

Lavado de envases

Preparaci´on de l´ıquido de gobierno

Cerrado

Embalado

Este desbalance provoca que exista demasiado tiempo improductivo en estas c´elulas de producci´on, por lo que se propone las siguientes estrategias de mejora:

**4.5.1. Asignaci ´on**

**on de mano de obra para balanceo de l´ıneas de producci ´**

Las c´elulas donde se puede cambiar la asignaci´on de mano de obra sin que afecte el proceso productivo son la de lavado de envases y la de embalaje de producto terminado. En las otras c´elulas no puede hacerse ya que el personal que est´a trabajando

***ICM 4-70 ESPOL***

actualmente cumple funciones espec´ıﬁcas y retirar personal afectar´ıa gravemente el desempe˜no de ´estas celdas.

Para la c´elula de lavado de envases se cuenta actualmente con 3 personas, de los datos observados en la Figura 4.1 se observa que la c´elula tiene un 52 % aproximadamente de tiempos improductivos, por lo que se propone las siguientes mejoras:

Trabajo de la c´elula sin tiempos muertos, esto es posible si se cambia la forma de trabajo para almacenar los envases lavados hasta que el producto est´e listo para ser envasado. Para esto se asign´o en la planta un espacio que puede almacenar 4 pallets de envase lavado. Cabe indicar que se analiz´o esta decisi´on con el comit´e HACCP de la empresa y no existe problema debido a que todos los envases vienen en cart´on por lo que una vez lavados se almacenan en el mismo cart´on.

Reducci´on de una persona de mano de obra directa. El tiempo improductivo con el nuevo m´etodo de planiﬁcaci´on equivale aproximadamente al 52 % por lo que podr´ıan retirarse dos personas sin embargo hay que tomar en cuenta que ahora el tiempo incrementa ya que se tiene que guardar el envase lavado en las cajas de cart´on y transportar estas cajas al espacio destinado para el almacenamiento.

Para la c´elula embalaje se cuenta actualmente con 3 personas, de los datos observados en la Figura 4.1 se observa que la c´elula tiene un 69 % aproximadamente de tiempos improductivos, por lo que se propone la siguiente mejora:

Reducci´on de dos personas de mano de obra directa. Esta reducci´on no causa problemas debido a que si la persona que se queda en la c´elula se retrasa en el embalaje se acumula el producto antes de pasar a bodega pero al ser la ´ultima operaci´on no afecta al secuenciamiento planiﬁcado.

**4.5.2. Cambio en el m´etodo de trabajo**

Espec´ıﬁcamente para la c´elula de preparaci´on de l´ıquido de gobierno se propone la siguiente mejora

Uniﬁcar la formulaci´on de l´ıquido de gobierno para todos los productos, con esto se prepara el l´ıquido de gobierno al inicio del turno y se lo almacena en

***ICM 4-71 ESPOL***

los tanques previstos para el efecto. De ´esta forma la persona de ´esta c´elula de producci´on trabajar´ıa aproximadamente 3 horas y luego pasar´ıa a otras c´elulas.

**4.6. Implementaci ´**

**on en la Empresa ABC**

El proceso de implementaci´on en la Empresa ABC consisti´o en planiﬁcar usando el algoritmo propuesto y se lo empez´o a usar a partir de la tercera semana luego de la ﬁnalizaci´on del desarrollo del mismo y con el balanceo de las c´elulas de producci´on. Con esto se pudo recopilar la informaci´on de 3 semanas con lo cual se obtiene la siguiente informaci´on.

**4.6.1. Tiempo de Para en las celdas de producci ´on**

En las tres semanas se recopil´o la informaci´on de para de cada celda de producci´on de forma similar con la que se obtuvo la informaci´on de la secci´on 2. Se presenta en la Tabla 4.6 el tiempo de para promedio diario.

Tabla 4.6: *Tiemp*o *d*e *Par*a *promedi*o *diari*o

|  |  |
| --- | --- |
| Causa | Tiempo (min) |
| Secuenciaci´on de Producci´on | 7,1 |
| Da˜no de m´aquina | 39,8 |
| Falta de personal | 12,2 |
| Otros | 20,3 |
| Total | 79,4 |

Se puede observar lo siguiente:

El mal secuenciamiento de producci´on paso a ser la causa menor de para de no planiﬁcada en las celdas de producci´on. El tiempo actual de 7,1 minutos indica que todav´ıa existe una mejor soluci´on pero para objetos pr´acticos el algoritmo desarrollado es excelente.

La falta de personal tambi´en disminuy´o debido a que actualmente se tiene dos personas ﬂotantes (provenientes del balanceo de las l´ıneas de producci´on) por lo que si falta una persona inmediatamente pasan a sustituirla.

***ICM 4-72 ESPOL ´***

**4.6.2. Horas extra pagadas no planiﬁcadas**

De igual forma que el indicador calculado en la secci´on 2, se procede a calcular el nuevo valor del indicador con lo que se tiene los datos de la Tabla 4.7. Cabe indicar que como se contaba ´

unicamente con datos de 3 semanas, se hizo una regla de tres para obtener el valor de 4 semanas y poder hacer el comparativo:

Tabla 4.7: *Hora*s *extr*a *n*o *planiﬁcada*s *po*r *me*s

|  |  |  |  |
| --- | --- | --- | --- |
| Mes | Horas extras 50 % | Horas extras 100 % | Total |
| Noviembre | 196 | 72 | 268 |
| TOTAL |  |  | 52715 |

Se puede observar que se tiene un valor de $268,00 USD en horas extras para el mes, lo cual da un valor anual aproximado de $7.380,00 USD el cual frente a los $150.000,00 USD que la empresa pagaba anteriormente representa un ahorro de $143.000,00 USD al a˜no.

**4.7. Conclusiones y Recomendaciones**

**4.7.1. Conclusiones**

Con respecto al objetivo principal de la investigaci´on, es decir la obtenci´on de un sistema que resuelva el problema de estudio se encuentra que:

Se desarroll´o un algoritmo gen´etico que resuelve problemas de la familia JSS debido a la facilidad de implementaci´on y ejecuci´on de los gen´eticos. Para la obtenci´on de los par´ametros del algoritmo de us´o el problema est´andar FT10 y se ﬁjaron valores iniciales obtenidos de la bibliograf´ıa, se modiﬁcaron uno a uno cada par´ametro y se guardaban los que permit´ıan obtener el mejor makespan para el problema en menci´on. A partir de esto se obtuvo los siguientes valores de calibraci´on:

Poblaci´on: 100

•

* N´umero de generaciones: 5000
* Probabilidad de mutaci´on: 0,091

***ICM 4-73 ESPOL***

* Probabilidad de cruce: 0,60
* Probabilidad de reemplazo: 0,86
* Semilla para generar n´umeros aleatorios: 99
* Mejor combinaci´on de operadores: operador de cruce PMX y operador de mutaci´on: mutaci´on combinada.

En el algoritmo planteado no fue necesario implementar una etapa de factibilizaci´on, debido a que la poblaci´on inicial se genera con secuencias factibles y los operadores implementados garantizan secuencias leg´ıtimas. Sin embargo, para futuras investigaciones, es posible considerar alternativas de soluci´on infactibles lo cual justiﬁcar´ıa una etapa de factibilizaci´on y permitir´ıa contar con un espacio de soluciones mayor.

Se dise˜no el algoritmo de forma que sea compacto en cuanto a recursos usados, para esto se trabaj´o sobre la representaci´on y los mecanismos de cruce y mutaci´on utilizados. Gracias a esto se utiliza poca memoria y el tiempo de ejecuci´on es razonable comparado con otros m´etodos existentes.

El algoritmo desarrollado en el presente proyecto cumple el objetivo propuesto inicialmente, el cual era obtener un mejor plan ﬁnito de producci´on en un tiempo menor al que se requer´ıa en el m´etodo de planiﬁcaci´on tradicional. En uno de los casos a resolver se pudo comparar que el algoritmo propuesto disminuy´o el makespan en 60 % del valor que se obten´ıa mediante la forma tradicional de planiﬁcaci´on. El tiempo requerido con el algoritmo gen´etico propuesto oscila de dos a tres minutos, mientras que anteriormente se tardaba dos d´ıas. Cabe recalcar que ´este valor es para uno de los casos propuestos en los que se pudo comparar los resultados obtenidos por los dos m´etodos.

La r´apida ejecuci´on del algoritmo permitir´a al planiﬁcador de la producci´on, realizar varios escenarios factibles y analizarlos con las particularidades que ocurren cada d´ıa de forma que cuando el desarrollo se convierte en una herramienta cuando hay imprevistos.

Con la implementaci´on del algoritmo gen´etico se espera obtener un ahorro de $143.000,00 USD al a˜no en reducci´on de horas extras al personal de la planta.

***ICM 4-74 ESPOL***

Mensualmente se debe monitorear este indicador para veriﬁcar que se ´este usando el m´etodo de planiﬁcaci´on propuesto.

El algoritmo presenta un 3 % de error medio en la resoluci´on de los problemas est´andar: FT20, FT10, LA24, LA27 y FT6. El tiempo promedio de ejecuci´on es de 2 minutos 47 segundos. Se puede concluir que a comparaci´on de los m´etodos metaheur´ısticos modernos con errores menores al 1 % y tiempos que oscilan entre 32 a 47 segundos, el algoritmo gen´etico desarrollado es muy bueno para la empresa de inter´es pero en el campo te´orico est´a muy por debajo del desempe˜no de los m´etodos actuales.

El algoritmo ha sido desarrollado en C++ usando como base el framework de algoritmos gen´eticos GALIB desarrollado por Matthew Wall en el Instituto Tecnol´ogico de Massachussets (MIT) [1]. Los experimentos han sido ejecutados en una LAPTOP Core 2 DUO T5500, 1.66Ghz, con una memoria RAM de 2.0 GB, bajo Windows XP Service Pack 3.

**4.7.2. Recomendaciones**

El algoritmo desarrollado no contempla la representaci´on gr´aﬁca de la soluci´on, a futuro se puede incluir el algoritmo en una aplicaci´on de entorno Windows que facilite la operaci´on del personal de la planta.

Otra recomendaci´on que se propone es extender al algoritmo desarrollado para crear un algoritmo h´ıbrido; de la bibliograf´ıa revisada se puede construir un algoritmo gen´etico que incluya mecanismos de la b´usqueda tab´u a ﬁn de mejorar su desempe˜no. Otra opci´on es el procesamiento paralelo, el cual es la ejecuci´on simult´anea de instrucciones en una computadora. La recomendaci´on es extender el algoritmo desarrollado para transformarlo en un algoritmo gen´etico de procesamiento en paralelo. El m´etodo m´as simple de paralelizar un AG se llama paralelizaci´on global; en donde existe una sola poblaci´on al igual que en el AG convencional, pero la evaluaci´on de los individuos y los operadores gen´eticos se paralelizan de forma expl´ıcita.

El algoritmo desarrollado optimiza la secuenciaci´on diaria de producci´on de la

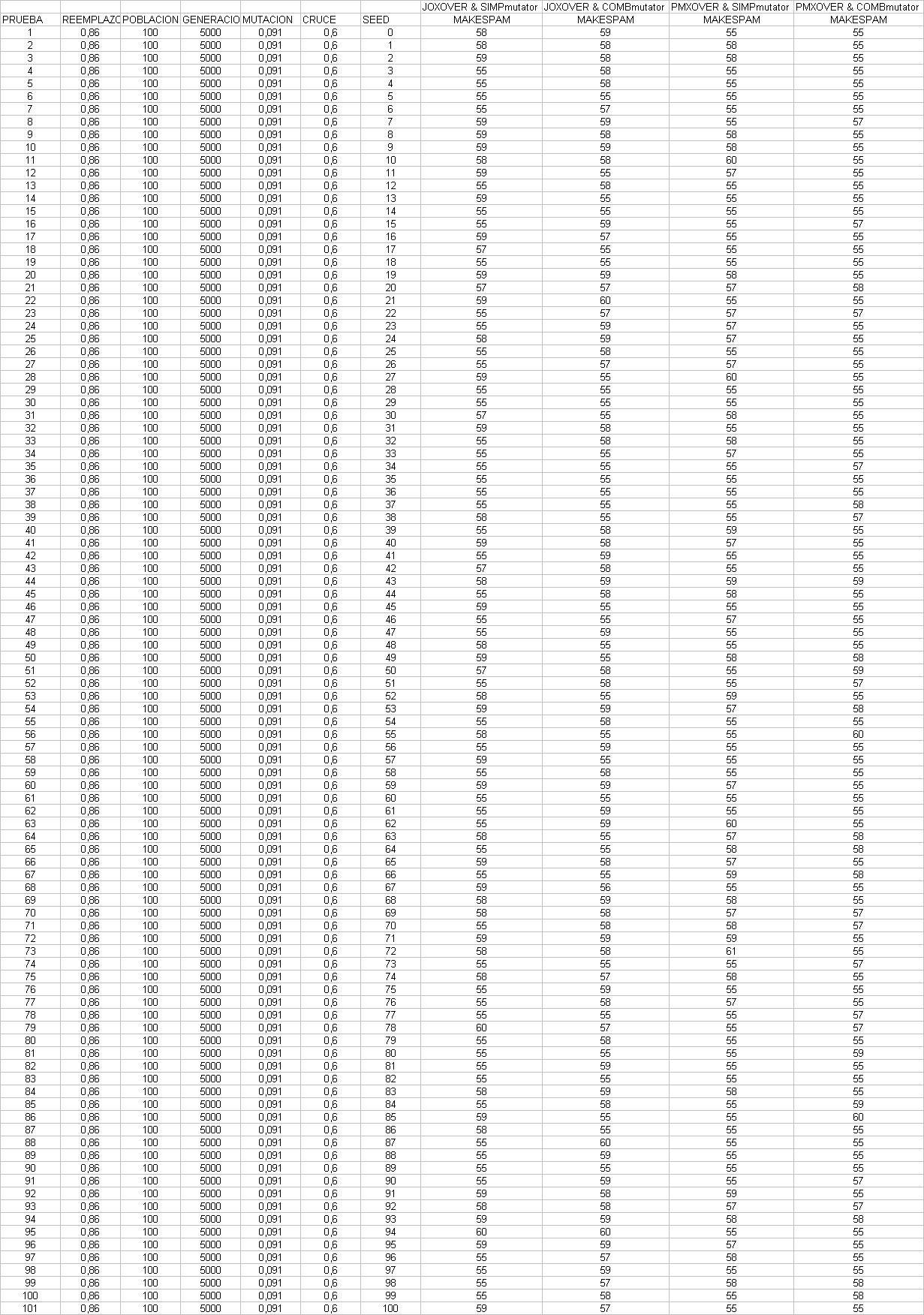
***ICM 4-75 ESPOL***

planta. Se ve a futuro la necesidad de extender el algoritmo para que permita realizar la secuenciaci´on de producci´on de per´ıodos m´as largos de tiempo. Para ello es necesario incluir restricciones como: la rotaci´on de personal, las fechas y cantidades de entrega de materia prima, la planiﬁcaci´on de mantenimiento de maquinaria, entre otras.

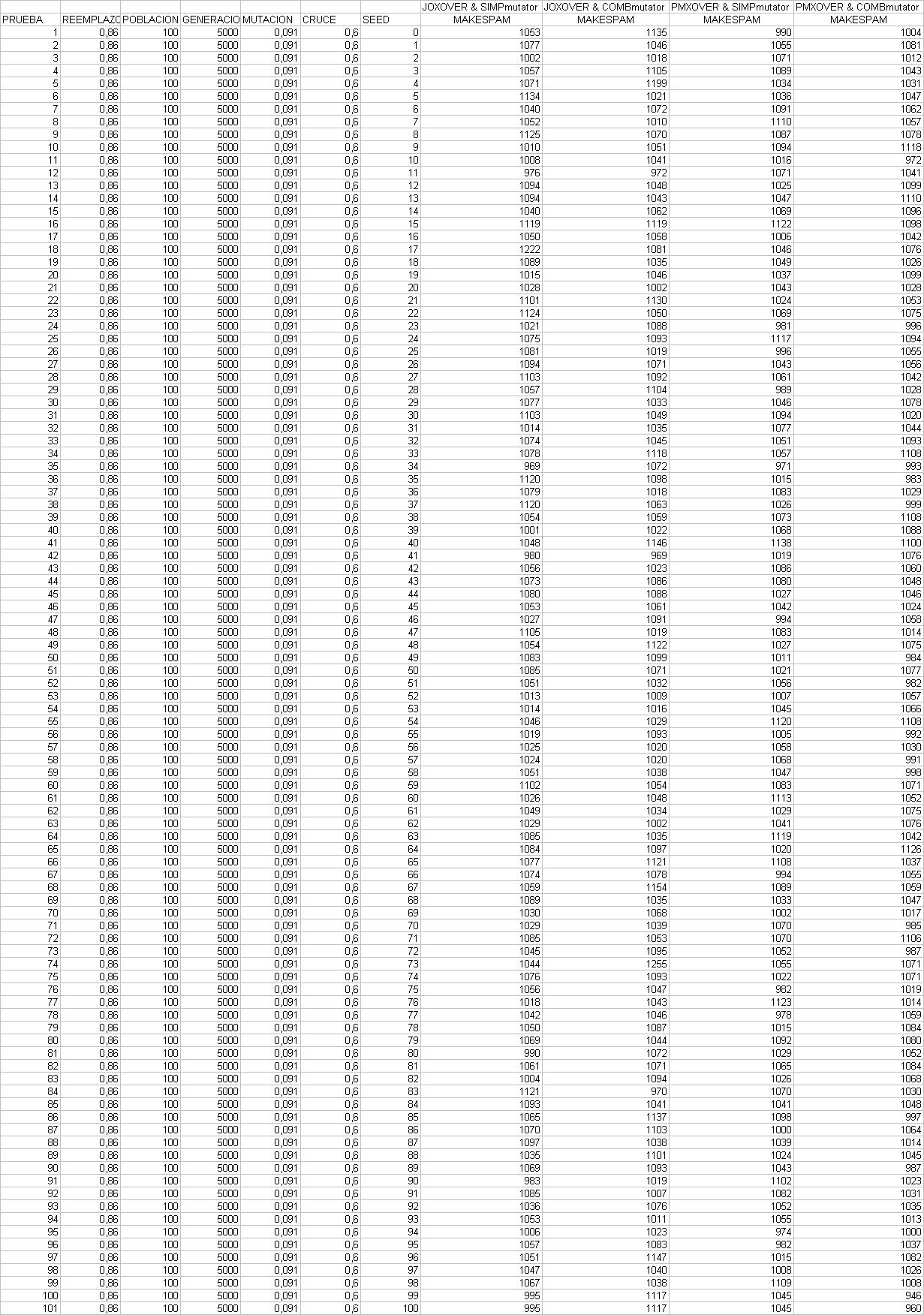
Existen diversos objetivos que pueden ser evaluados al buscar una secuencia ´

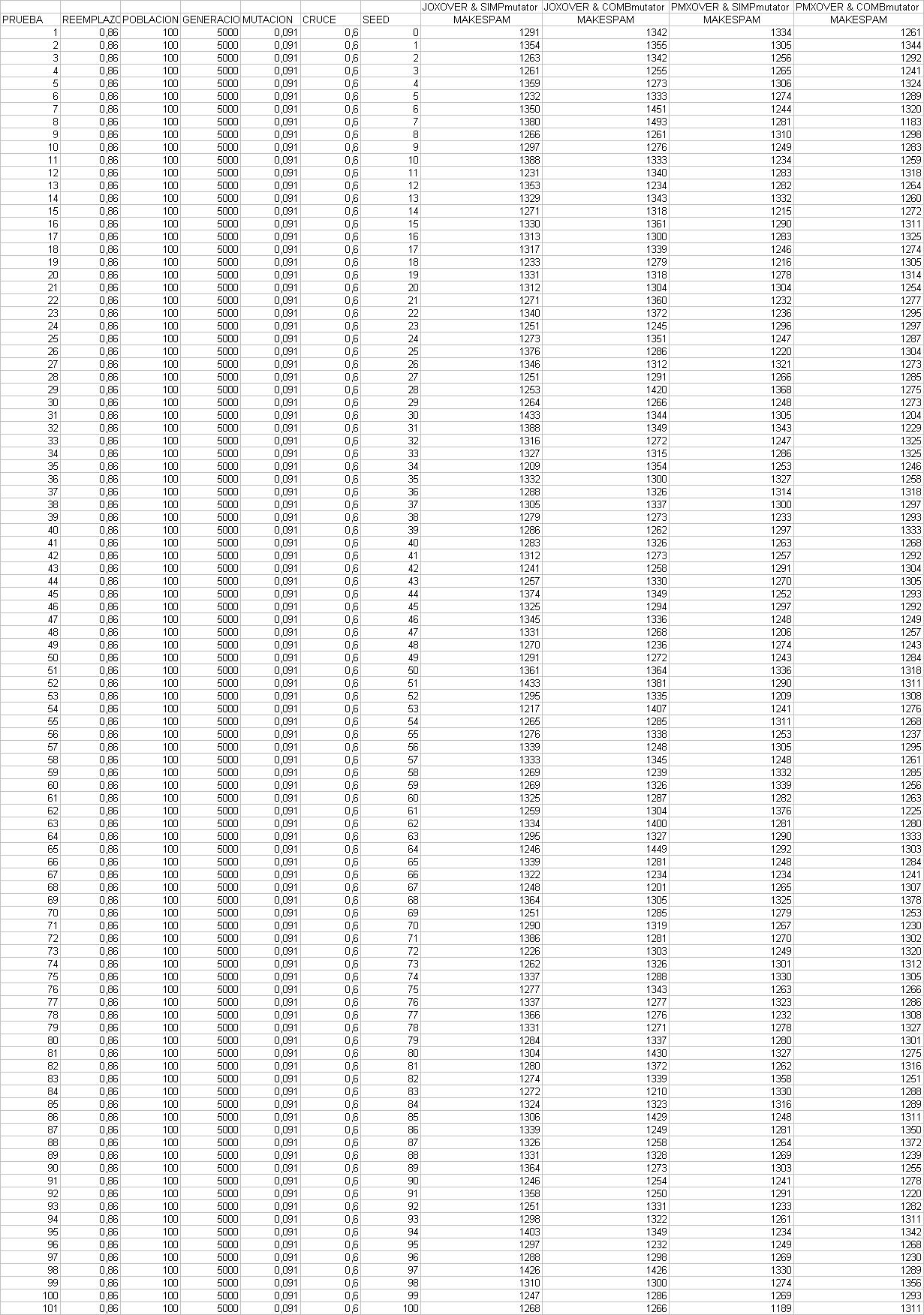
optima, como por ejemplo: cumplir con las fechas de entrega, minimizar la tardanza de trabajos, minimizar horas extras, maximizar la utilizaci´on de m´aquinas, minimizar el tiempo imporductivo, o minimizar inventario de trabajo en proceso. La selecci´on del objetivo a optimizar depende de la necesidad del usuario, en este trabajo se minimiz´o el tiempo de procesamiento total (Cmax). Sin embargo se puede estudiar la posibilidad de mejorar el algoritmo para que pueda realizar optimizaci´on multi-objetivo.

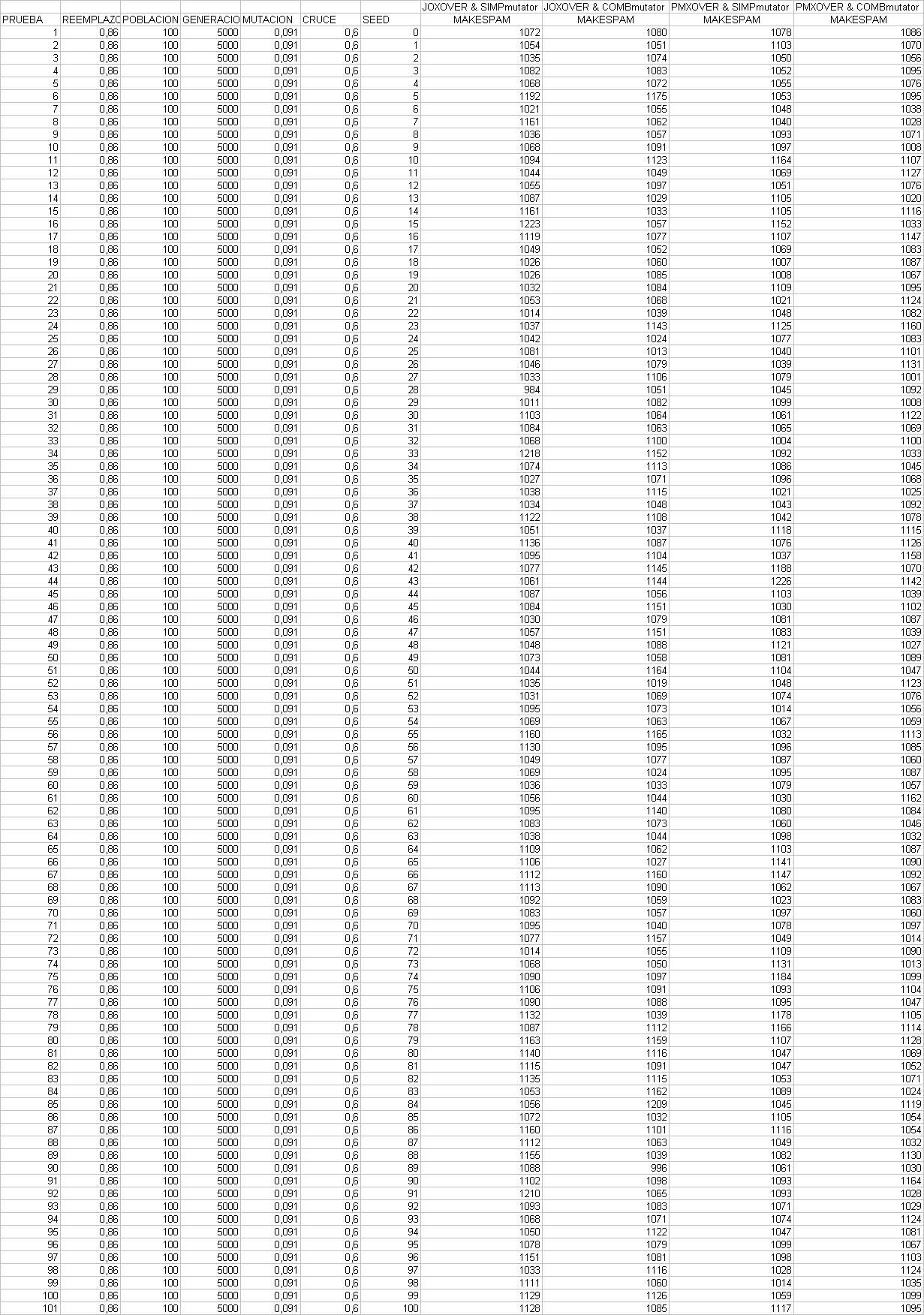
**ANEXO**

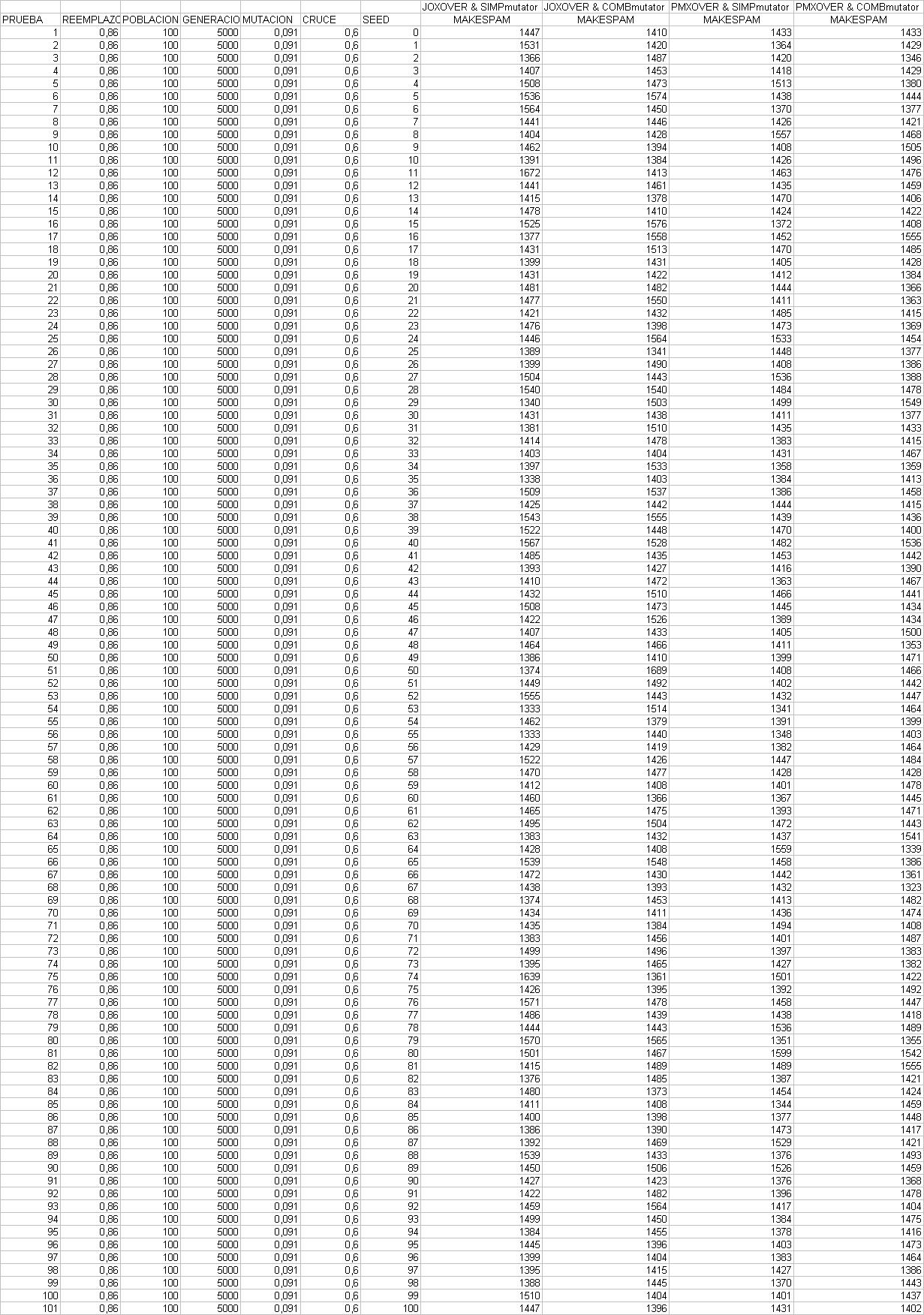


Anexo i: Pruebas de la instancia FT6 Anexo ii: Pruebas de la instancia FT10 Anexo iii: Pruebas de la instancia FT20 Anexo iv: Pruebas de la instancia LA24 Anexo v: Pruebas de la instancia LA27









**BIBLIOGRAFIA**

[1] Instituto tecnol´ogico de massachussets. Web site: http://lancet.mit.edu/ga/, 2009.

[2] Or-library librer´ıa de investigaci´on de operaciones. Web site: http://people.brunel.ac.uk/ mastjjb/jeb/orlib/jobshopinfo.html, 2009.

[3] J. Balas. The shifting bottleneck procedure for job shop scheduling. pags. 394–401, Management Science Vol 34, USA, 1998.

[4] C. Blum. Meta-heuristics in combinatorial optimization: Overview and conceptual comparison. In *An*n *Discret*e *Math.*, pags. 269–308, ACM Comput Surv, 2003.

[5] C. Blum. Hybrid metaheuristics. studies in computational intelligence. In *vo*l *114*, pags. 17–37, 2008.

[6] M. Bramlette. Initialization, mutation and selection methods in genetic algorithms for function optimization. In *Proceeding*s *o*f *th*e *Fourt*h *Internationa*l *Conferenc*e *o*n *Geneti*c *Algorithms*, pags. 100–107, 1991.

[7] P. Brucker. Complex scheduling. springer science and business. pags. 180–210, Media Inc. New York, 2006.

[8] P. Brucker. Scheduling algorithms ﬁfht edition. springer science and business. pags. 170–190, Media Inc. New York, 2007.

[9] D. F. E.S. Roscoe. Organization for production. In *ﬁft*h *edition*, *Richar*d *D*. *Irwi*n *Inc*. *Homewoo*d *Illinois*, 15 pags., 1971.

[10] J. Herrmann. Handbook of production scheduling. In *Handboo*k *o*f *Productio*n *Scheduling*, pags. 2–20, Springer Science and Business Media Inc., New York, 2006.

[11] C. J.F. Blackstone, J.H. Spencer. Apics dictionary. In *America*n *Productio*n *an*d *Inventor*y *Contro*l *Society*, pags 17-35, Virginia, 2001.

[12] R. L. LaForge. Manufacturing scheduling and supply chain integration: a survey of current practice. In *America*n *Productio*n *an*d *Inventor*y *Contro*l *Society*, pags. 1–16, Falls Church, Virginia, 1998.

[13] J. Lenstra. Computational complexity of discrete optimization problems. In *An*n *Discret*e *Math.*, pags. 121–140, 1979.

***ICM 82 ESPOL***

[14] M. Mitchell. An introduction to genetic algorithms complex adaptive systems. mit press. In *Massachusett*s *Institut*e *o*f *Techology*. *Cambridge*, *Massachusetts*, 12-69 p, 1996.

[15] O. Wight. Production and inventory management in the computer age. In *Va*n *Nostran*d *Reinhol*d *Company*, pag 24, New York, 1984.

[16] F. Xhafa. Metaheuristics for scheduling in industrial and manufacturing applications.