2M11.2787.8

University Of Montreal

ALLOTMENT OF AIRCRAFT SPARE PARTS USING GENETIC ALGORITHMS

A thesis submitted to the

Department of Computer Science and
Operations Research
Faculty of Arts and Science

By

Pascale Batchoun

in partial fulfillment of the requirements
for the degree of

Master of Science (M.Sc.)

Computer Science and Operations Research

July 2000

© Pascale Batchoun, 2000

Faculté des Études Supérieures
Grade octroyé
à compter du
OCT 5 2000
Université de Montréal

QA
76
U54
2000
v.037

ALLOTMENT OF AIRCRAFT SPARE PARTS USING GENETIC ALGORITHMS

A thesis submitted to the

Department of Computer Science and
Operations Research
Faculty of Arts and Sciences

By

Pascale Batchoun

Will be evaluated by the following jury members:

President: Patrice Marcotte
Director: Jacques Ferland
Co-director: Robert Cléroux
Member: Michael Florian

ABSTRACT

ALLOTMENT OF AIRCRAFT SPARE PARTS USING GENETIC ALGORITHMS

by Pascale Batchoun

This thesis attempts to determine the optimal distribution of aircraft parts used as spares for replacement of defective parts on-board of a departing flight. In order to minimize the cost of delay caused by unexpected failures, Genetic algorithms (GAs) are used to distribute the initial quantity of parts among the airports. GAs are a class of adaptive search procedures, and distinguish themselves from other optimization techniques by the use of concepts from population genetics to guide the search. Problem-specific knowledge is incorporated into the problem and efficient parameters are identified and tested for the task of optimizing the distribution of parts.

RÉSUMÉ

Quand une pièce d'avion devient défectueuse, il faut la remplacer par une autre en réserve avant de permettre le décollage. Dans le cas où il n'y a pas de pièce de rechange sur place, le vol est retardé jusqu'à l'arrivée d'un autre vol en provenance d'un endroit où une telle pièce est disponible. De plus, la pièce défectueuse est acheminée à l'usine d'entretien pour réparation. Si l'usine dispose d'une pièce de rechange en bon état, elle en retourne une à la station immédiatement. Dans le cas contraire, la pièce défectueuse est réparée puis retournée à cet endroit. Le but est de réduire les coûts de délai dus aux pannes imprévisibles en redistribuant adéquatement les pièces de rechange aux divers aéroports du réseau et à l'usine d'entretien pour permettre le remplacement des pièces défectueuses. Dans ce mémoire, nous résolvons ce problème de redistribution optimale d'un nombre spécifié de pièces de rechange à l'aide de méthodes de recherche adaptative de type génétiques.

Le réseau d'Air Canada couvre 60 aéroports avec plus de 3 300 vols hebdomadaires. La flotte est composée de 190 appareils de 8 types différents. Plus de 1 250 types de pièces de rechange différents doivent être redistribués. Le nombre de pièces de rechange disponibles pour chacun des types varie entre 1 et 11. Puisque la redistribution des pièces d'un type est indépendante de celles des autres, il s'ensuit que chaque problème peut être résolu indépendamment des autres.

Considérons un type spécifique de pièce. Une solution du problème correspondant est une redistribution des pièces de rechange aux aéroports du réseau et à l'usine d'entretien. Elle est optimale si son coût de délai dû aux pannes imprévisibles est minimale. Nous faisons l'hypothèse que le nombre moyen de

pannes pour chaque pièce d'un type spécifique est distribué selon une loi de Poisson. Son paramètre est estimé à partir des données historiques des pannes en considérant les type d'avion où nous la retrouvons et les heures de vols complétées par la flotte. Le coût moyen de délai dû à une panne est estimé en référence à l'horaire courant des vols. Il faut rappeler que le délai dû à une panne imprévisible de la pièce à un aéroport donné est fonction des arrivées des vols en provenance d'origines où une telle pièce est disponible. Lors d'une panne, le nombre de pièces de rechange disponibles à un aéroport peut être différent de celui prévu dans la redistribution initiale. En effet d'autres pannes peuvent avoir entraîné l'utilisation de certaines pièces de rechange et l'acheminement des pièces défectueuses à l'usine pour réparation. Ainsi, la probabilité d'avoir une pièce de rechange lors d'une panne dépend également du temps de réparation à l'usine si une pièce de rechange en bon état n'est pas disponible pour acheminement immédiat à l'aéroport. Il s'ensuit que l'évaluation du coût d'une solution exige beaucoup d'effort de calcul principalement dû au fait que tout l'horaire courant des vols doit être pris en compte.

Il serait difficile d'utiliser des méthodes de résolution conventionnelle basées sur des algorithmes exacts puisque la fonction économique est très complexe et requiert beaucoup d'effort de calcul. De plus, celle-ci ne comporte pas de structure particulière qui pourrait être exploitée afin d'évaluer facilement l'effet de modifier légèrement une solution. Lorsque le nombre de pièces de rechange est petit (c'est-à-dire égal à un ou deux), une méthode de recherche exhaustive peut être utilisée pour déterminer la solution optimale puisque l'espace des solutions est réduit. Par contre, l'espace des solutions augmente rapidement avec le nombre de pièces de rechange et peut atteindre une dimension de plus de quatre millions de solutions. Dans ces cas, la recherche exhaustive devient beaucoup trop coûteuse et nous devons avoir recours à des méthodes heuristiques. C'est ce qui nous a conduit à utiliser des méthodes de type génétique pour résoudre notre problème de redistribution des pièces de rechange.

Les méthodes génétiques sont des techniques d'optimisation globale permettant de contourner certaines difficultés rencontrées avec les méthodes de recherche locale où en particulier le risque d'identifier une solution qui n'est que localement optimale est plus grand. Ces méthodes utilisent des populations de solutions qui héritent de certaines propriétés des populations antérieures. Elles permettent une recherche plus exhaustive de l'espace des solutions. Dans un algorithme génétique, la façon de représenter les solutions est un élément important de la méthode pour faciliter la génération des populations. Les solutions sont dénotées chromosomes, et à chaque itération (ou génération) nous disposons d'une population de ceux-ci. Une nouvelle population est générée en incorporant certaines caractéristiques des populations précédentes. Par analogie avec le phénomène de l'évolution et de la sélection naturelle, nous voulons favoriser la transmission des meilleurs gènes en sélectionnant les meilleurs chromosomes pour générer la prochaine population. Ainsi, il faut évaluer le coût de délai de chaque chromosome de la population courante afin de pouvoir choisir les meilleurs pour se reproduire. Une fois sélectionnés, ils sont regroupés par paires pour produire des chromosomes enfants à l'aide de divers mécanismes de croisement qui donnent lieu à diverses variantes de la méthode. Un opérateur de mutation permet de modifier légèrement ces chromosomes enfants à l'aide de perturbations aléatoires pour mieux explorer tout l'espace des solutions. La dimension des populations successives est maintenue constante en sélectionnant certains chromosomes enfants et certains de la population courante qui sont à leur tour utilisés pour générer la prochaine population. Cette procédure itérative est répétée pour un nombre fini de fois, et la meilleure solution rencontrée est retenue. Certaines variantes de la procédure s'arrêtent lorsque les chromosomes de la population sont presque identiques.

Plusieurs paramètres doivent être ajustés pour obtenir une variante performante pour résoudre le problème. Il faut d'abord choisir les solutions (chromosomes) qui constituent la population initiale. Pour notre problème de redistribution de pièces de rechange, il semble naturel de croire que le coût de délai est plus petit pour les solutions où davantage de pièces de rechange sont disponibles aux aéroports où la fréquence des départs est plus grande. Nous avons vérifié cette hypothèse à l'aide d'une série de tests préliminaires. Ainsi, certaines solutions exploitant la structure du problème sont générées où la redistribution des pièces aux divers aéroports est proportionnelle à la fréquence relative des départs. De plus, nous avons analysé quelle devrait être la proportion de telles solutions présentes dans la population initiale. Des tests numériques indiquent qu'une proportion égale à 50% permet d'obtenir des meilleures résultats qu'avec des proportions de 20% ou 80%.

La dimension de la population à chaque génération est aussi un paramètre important. Elle est en général proportionnelle à la dimension du problème, ce qui correspond au nombre de pièces de rechange à redistribuer dans notre cas. Les meilleurs résultats sont obtenus avec des populations de dimension égale à cinq fois le nombre de pièces à redistribuer.

Dans notre problème la représentation des solutions sous forme de chromosomes consiste à associer à chaque solution un vecteur de dimension égale au nombre de pièces à redistribuer où chaque élément correspond à un aéroport où une pièce est affectée.

La sélection des chromosomes de la population qui sont utilisés pour la reproduction est complétée selon un processus aléatoire biaisé en faveur de ceux qui sont plus performants. Alors plus un chromosome a un coût de délai faible plus il a de chances d'être sélectionné. Ainsi, le même chromosome peut être sélectionné à plusieurs reprises. Les chromosomes sélectionnés sont regroupés par paires pour générer deux chromosomes enfants selon un processus de croisement. Les résultats numériques indiquent que des meilleurs résultats sont obtenus avec un croisement de type *uniforme* plutôt qu'avec un croisement de type *1-point*. L'opérateur de mutation permet de modifier chaque chromosome enfant en remplaçant avec une très faible probabilité chaque élément du vecteur correspondant par un autre aéroport choisi aléatoirement.

La nouvelle population où seront choisis les prochains chromosomes pour reproduction est constituée des parents et des chromosomes enfants. Les résultats numériques indiquent qu'il vaut mieux conserver 10% des meilleurs parents de la population précédente et sélectionner les autres avec le processus de sélection aléatoire biaisé précédent.

La structure de notre problème, la représentation des solutions et les opérateurs utilisés permettent d'assurer que les chromosomes enfants générés satisfont les contraintes de notre problème. Pour des problèmes différents associés à des pièces différentes et appartenant à des types d'avion différents, les résultats numériques indiquent que le coût des solutions générées diffère d'au plus 1.3% de celui de la meilleure solution obtenue.

Dans le dernier chapitre, nous comparons notre modèle avec METRIC (Multi-Echelon Technique For Recoverable Item Control) developpé par la gestion d'inventaires d'éléments reparables. Nous soulignons les similarités et les différences entre ces deux modèles. Également, nous résumons brièvement le modèle RAPS de Tedone (1989) qui est utilisé chez American Airlines.

Finalement, il importe de noter que l'algorithme requiert beaucoup de temps de calcul. Or, ceci est un moindre mal en considérant qu'originalement l'algorithme était destiné à être utilisé mensuellement pour redistribuer les pièces lors des changements d'horaire de vols. Il faut également rappeler que le plus gros de

l'effort de calcul est consacré à l'évaluation des coûts de délai. Ainsi, si l'algorithme devait être utilisé sur une base plus fréquente, ou en temps réel, il y aurait lieu de remplacer l'évaluation des coûts de délai par des approximations plus faciles à obtenir pour accélérer le processus de résolution. Il serait également intéressant de poursuivre les recherches afin d'identifier des opérateurs de croisements mieux adaptés et tirant avantage de la structure spécifique du problème.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

When a repairable item on an aircraft becomes defective, it is removed and replaced by another item from the spare stock. The defective part is then sent to the maintenance shop for repair. Should the station not have a spare part in stock, the aircraft will remain on ground and will be delayed until an incoming flight brings a replacement part. In this context, a repairable item represents any aircraft part from a small electronic component to a whole engine. Spare inventory is placed at line stations to decrease time delays due to unanticipated failures, and is placed at the maintenance shop to quickly replenish remote station allotments while their parts are being repaired.

The objective of the study is to answer the following question: given the initial provisioning of repairable spare parts, what appropriate amount should be allocated at the different stations around the world and at the maintenance shop to adequately and economically keep the airline fleet flying with minimal delay costs?

This work attempts to determine the optimal distribution of parts using a class of adaptive search procedures called genetic algorithms (GAs). The class of GAs distinguish themselves from other optimization techniques by the use of concepts from population genetics to guide the search.

In the next chapter, the problem is described in details, presenting the process of replenishment of unserviceable parts and the role of the repair shop. Then the problem formulation is introduced in chapter 2, as well as the failure process of parts and the evaluation of the delay for a given solution. Like other classes of algorithms, GAs include several parameters and strategies leading to several variants. In Chapter 3, we describe the Genetic Algorithm implementation and its variations. Chapter 4 includes the experiments to identify efficient parameters for the task of optimizing the distribution of parts. In the last chapter, our model is compared to the METRIC model, a well-known mathematical model for spares inventory, and is also compared to other recent researches.

GLOSSARY

Part. A repairable equipment on-board of an aircraft

Serviceable Part. A part that is fully functional.

Unserviceable Part. A part that is defective, non-functional.

Station. An airport that is part of the airline's network.

Maintenance Capability. A station is said to have maintenance capability for a specific part if there exist at all times maintenance personnel that can check the part when it becomes defective, remove it and replace it by a serviceable one. A station can have maintenance capability for a specific part and not for another.

Removal. The failure and removal of a part that becomes defective.

AOG. Aircraft On Ground: An aircraft that is unserviceable due to a defective part.

MOT Regulations. Minister Of Transport Regulations

MTBR. Mean Time Between Removals is the expected flying hours between two removals.

Shop Float. The number of spare parts placed at the maintenance shop for replenishment purposes.

Transit Time. The time it takes to send an unserviceable part to the maintenance shop for repair plus the time for the shop to send back a serviceable one to the station.

Turn Around Time (TAT). Time required to repair a defective part at the shop.

Replenishment Time. Time required to receive a serviceable part at the station from the shop in replacement of its unserviceable one. It is equal to the Transit Time in the case where the shop has the part in stock. In the case where the shop float is nil, the replenishment time is equal to the Transit Time plus the Turn Around Time.

Chapter 1

PROBLEM DESCRIPTION

1.1 Background

Air Canada's network of flights covers a total of 60 destinations, with more than 3,300 flights per week. The fleet includes 190 aircraft of 8 different types. The Purchasing group at Air Canada determines the number of spare parts needed every time a new aircraft is purchased. The spares are then distributed among the airports that are covered by the flight schedule in order to replace defective parts onboard a departing flight. The spare parts are distributed in a way to minimize the delay caused by unexpected failure of parts, and it is based on historical data for failure of parts per station. However, there are 1,258 different types of parts to be distributed among the stations. Furthermore, every time the schedule changes, the allotment of parts has to be revisited since the allotment of spares is dependent on the frequency of departures per station, per aircraft type. Most of the work is done manually and parts are not always distributed in an optimal way. This thesis intends to automate the process of optimizing the distribution of 1,258 different types of spare parts among the stations in order to minimize the cost of delay.

1.2 Nature of parts

The essentiality code of a part determines how quickly a defective part should be replaced in order for the aircraft to become serviceable. There are three essentiality codes: If a part has an essentiality code 1, then the aircraft becomes in an unserviceable state should one of these parts become unserviceable. When a part of code 1 fails, the aircraft is grounded (AOG status) until the defective part is replaced by a functional one. If a part has an essentiality code 2, then the defective item should be replaced within the time interval specified in its qualifying conditions. For example a part might have to be replaced within 3 consecutive days following its failure, or the aircraft becomes unable to operate an overseas flight until the part is replaced. An essentiality code of 3 implies that the functionality of the aircraft is independent of the status of the part. Therefore, when a part with essentiality code 3 becomes unserviceable, the aircraft will remain in a serviceable state.

The aircraft parts that are considered in this study are of essentiality code 1. This type of parts is causing the aircraft to be in an unserviceable state should one of them becomes unserviceable. Therefore, the aircraft is grounded, and the departure of the flight is delayed until the defective part is replaced by a functional one. There are 1,258 different types of parts of essentiality code 1 for a total of 3,178 parts. The initial quantity of spares per part type varies between 1 and 11 spares and their distribution is illustrated in Figure 1.

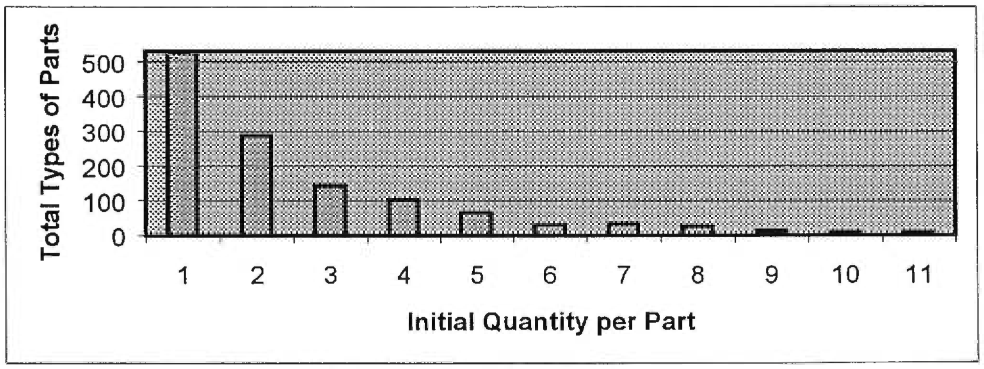

Figure 1: Quantity per Part Type

For example, there are 100 different types of parts having an initial quantity of 4, and 33 other types with an initial quantity of 7. Note that the functionality of one type is totally independent of another type, and therefore each type of parts yields a separate optimization problem.

1.3 Replenishment of unserviceable parts

When an unexpected failure of a part happens prior to a flight departure, the defective part is replaced from the station's stock should the departure station at the time of failure have a serviceable spare. Otherwise, if spare stock at the station is nil, the only way to receive a replacement part is through a scheduled flight arriving at that station, bringing a part from the spare stock of its originating station. In this case, the flight is delayed until a serviceable part becomes available at the station.

1.4 Repair Shop

When a part on an aircraft becomes defective, it is sent to the shop for repair. There is only one repair shop, located at the Dorval station (YUL). Repair times or Turn Around Times vary from one type of part to another. However, the average repair time per part is about 30 days. The shop keeps spare inventory in order to quickly replenish the station allotments. So when the shop receives a defective part, it sends immediately a serviceable one to the station, in replacement of the unserviceable part. If there is no spare available at the shop, the unserviceable part is repaired and sent back to its station. The shipping time of the part to the shop and back to its original station is called the Transit Time. Note that spare stock of the repair shop is the same as the one of Dorval station.

1.5 Pooled Items

"International Airlines Technical Pool" is an agreement between airlines to share specific parts at specific stations when parts fail. An airline is said to be the *provider* of a part at a specific station, when all the *participant* airlines of the agreement can borrow from the spare stock of the provider. For example, when a part of a *participant* airline becomes defective at a specific station, the *provider* airline is expected to provide a replacement part. There is a pool list that gets updated every two to three months, indicating the part being pooled, the station at which it is pooled, the provider of the part, and the list of participants.

When Air Canada is the provider of a part at a station, a minimum of one part is allocated by Air Canada at that station. When Air Canada is a participant for a part at a station, the airline doesn't allocate any parts of this type, since it is reliant on the provider.

1.6 Maintenance Ability

A station is said to have maintenance ability for a certain part, when there is maintenance staff at the station accredited to remove and replace the defective part on the aircraft. When there is no maintenance ability for a specific part at a station, the aircraft with a defective part remains on ground until a technician flies into the station along with a replacement part. Therefore, the stations are given nil allotment for parts with no maintenance ability. Note that there might be accredited personnel at a station to replace some type of parts but not all types.

Taking into account the pooling opportunities and the maintenance accreditations, the maximum number of stations where spare parts can be allotted (feasible stations) is reduced from 60 to 15 stations. Depending on the part type and on the aircraft types to which it belongs, the number of feasible stations varies between 10 and 15.

Chapter 2

PROBLEM FORMULATION

The mathematical formulation of the problem is quite straight forward, but the evaluation of the objective function is complex and time consuming. Since the notation required is quite involved, we summarize it in the following section.

2.1 Notations

N_j: The quantity of parts of type j to be distributed among all stations

S: The total number of stations that are covered by the airline's network of flights

A: Set of indices associated with all stations i or airports covered by the airline's network

P_j: Set of all stations i where Air Canada is the provider of the pool for part j

A_j: Set of all stations i where Air Canada is a participant to pool part j

M_j: Set of all stations i where there exists maintenance capability for part j

x_{ij}: The initial number of parts of type j allocated at station i

$x_{ij}(t)$: The number of spares of type j available at time of failure t at station i

$X_j = (x_{1j}, x_{2j}, .., x_{sj})$: A solution vector for parts of type j

$D(X_j)$: The expected cost of delay caused by unexpected failure of parts j. It is a function of the solution vector X_j

d_{ij}: The average cost of one delay per part failure of type j, for all flights departing from station i

δ_{ijk}: The delay in minutes that might be caused by failure of part j on-board flight k, departing from station i

$\delta_{ijk}(i_o)$: The delay δ_{ijk}, given that a replacement part was supplied by station i_o, on a flight from station i_o to station i

τ_j: Mean Time between Removals of part j

$Q(j,u)$: Quantity of parts j on board an aircraft of type u

$H(u)$: Total number of flying hours per week completed by all the aircrafts of type u

$F(i,u)$: Total frequencies of flights departing from station i per year, and operated by aircraft of type u

λ_j: Expected number of removals of part j per year

λ_{ij}: Expected number of removals of part j at station i per year

R_j: Replenishment time of part j

T_T: Transit Time

T_{Aj}: Turn Around Time of Part j

L: Service level or probability of replenishment of a serviceable part right after the unserviceable one is received at the repair shop

μ_{ij}: Expected number of removals of part j causing a delay at station i

2.2 Objective function and constraints

Consider a specific part type j, with initial quantity N_j on hand. Let S be the total number of stations operated by the airline: $S = |A|$. Let x_{ij} be the number of parts j allocated at stations $i \in A$. The problem is to distribute the N_j parts at the S stations in order to minimize the expected cost of delay $D(X_j)$ due to unexpected failures. The problem then is to determine the optimal solution $X_j = (x_{1j}, x_{2j}, .., x_{sj})$ to the following optimization problem:

$$Min \quad D(X_j)$$

Eq. 2- 1

$$s.\ t. \quad \begin{cases} \sum_{i=1}^{S} x_{ij} = N_j \\ x_{ij} \geq 1 \quad \forall\, i \in P_j \\ x_{ij} = 0 \quad \forall\, i \in A_j \\ x_{ij} = 0 \quad \forall\, i \in \overline{M}_j \end{cases}$$

$$P_{P_j} = \{i \in A \,/\, \textit{Air Canada is the provider of part j at station } i\}$$
$$P_{A_j} = \{i \in A \,/\, \textit{Air Canada is a participant to pool part j at station i}$$
$$M_j = \{i \in A \,/\, \exists \textit{ maintenance capability for part j at station } i\}$$
$$\overline{M}_j = A - M_j$$

Note that $1 \leq j \leq 1{,}258$ and $1 \leq N_j \leq 11$. Each type of part is a separate problem; i.e., there are 1,258 different optimization problems.

2.3 Evaluation of the cost of delay $D(X_j)$

A solution is a way of distributing or allotting spare parts at the stations and at the maintenance shop. A solution $X_j = (x_{1j}, x_{2j}, .., x_{sj})$ is said to be optimal if it induces the least cost of delay due to unexpected failures. In this section, we indicate how to determine the excepted cost of delay $D(X_j)$ given a specific allotment of spares X_j.

2.3.1 Average cost of delay per removal

The total time delay due to the failure of a part j prior to flight departure k includes the time to receive a serviceable part and the time to replace it on the aircraft. Since the replacement time of part j is constant regardless of the number of spares allocated at the station, it is not included in the time delay δ_{ijk}. Then if a station has a spare part to replace the unserviceable one at the time of failure, the time delay is nil. Otherwise, if there is no spare stock at the station, then a spare part has to be flown in through an airline's scheduled flight arriving at the station i, and originating at some other station i_o. Therefore in this case, the time delay δ_{ijk} is the time it takes to receive a part through this incoming flight.

The flights considered in this study belong to a one week flight schedule. Consider the flight k, $i \to i_d$, from the origin station i to a destination station i_d. To determine the time delay δ_{ijk} in the case where no spare part j is available at the time of failure on flight k, we take into account each flight $i_o \to i$, leaving from origin station i_o, within 24 hours after the scheduled departure time of flight k, such that station i_o can provide a spare part j. Denote $\delta_{ijk}(i_o)$ the time delay of flight k if the spare part j is provided at station i by an incoming flight at station i and originating from station i_o. To illustrate the evaluation process, consider the flights illustrated in Table 1.

k	i	i_d	$d_{i \to i_d}$	i_o	$a_{i_o \to i}$	$\delta_{ijk}(i_o)$
#870	KIN	YYZ	13:40	MBJ	18:05	265
#275	YYC	YEG	12:55	YYZ	15:28	153
				YWG	17:20	265
#556	YYZ	DCA	19:00	YUL	20:12	72
				BOS	20:55	115
				EWR	21:12	132

Table 1: Example for Replenishment of Parts

First, consider flight k, #870, from origin i, KIN, to destination i_d, YYZ. It is scheduled to leave from i at 13:40 GMT ($d_{i \to i_d}$ =13:40, the departure Greenwich time from station i of flight $i \to i_d$). Suppose that a part j breaks down on the aircraft servicing flight k, #870, and suppose also that $x_{ij}(t) = 0$ (the number of spare parts j at station i at time t = 13:40 is equal to 0). The only incoming flight to i, KIN, within 24 hours after 13:40 is coming from MBJ (flight MBJ $\rightarrow$ KIN), and it arrives at 18:05 GMT ($a_{i_o \rightarrow i}$ =18:05, the arrival time at i of flight $i_o \rightarrow i$). Hence, flight $i_o \rightarrow i$ arrives 265 minutes after the scheduled departure time of flight $i \rightarrow i_d$. Now, if station MBJ has spare part j available in stock at time t (i.e. $x_{i_o j}(t) > 0$), then flight KIN $\rightarrow$ YYZ has a 265 minutes delay: i.e., in this case, $x_{ij}(t) = 0$, $x_{i_o j}(t) > 0$, and

$$\delta_{ijk} = \delta_{ijk}(i_o) = a_{i_o \rightarrow i} - d_{i \rightarrow i_d} = 265 \text{ minutes.}$$

On the other hand, if MBJ has no part in stock at time t (i.e. $x_{i_o j}(t) = 0$), and since no other incoming flight in KIN can provide a spare part j within the next 24 hours, then the time delay is assumed to be

$$\delta_{ijk} = 24 \times 60 = 1440 \text{ minutes.}$$

In the second example, the flight k, #275, is such that there exist two incoming flights in its origin i, YYC, within 24 hours of its scheduled departure time. Suppose that a part j breaks down on the aircraft servicing flight k, and suppose that no spare part j is available at station i at time $t = 12:55$ (i.e. $x_{ij}(t) = 0$). If station i_{o1}, YYZ, has a spare part j in stock, (i.e. $x_{i_{o1} j}(t) > 0$), then a spare part j can be flown in i through flight YYZ $\rightarrow$ YYC. Hence

$$\delta_{ijk} = \delta_{ijk}(i_{o1}) = a_{i_{o1} \rightarrow i} - d_{i \rightarrow i_d} = 153 \text{ minutes.}$$

Now, if station i_{o1}, YYZ, has no spare part j in stock, but station i_{o2}, YWG, does, then

$$\delta_{ijk} = \delta_{ijk}(i_{o2}) = a_{i_o \to i} - d_{i \to i_d} = 265 \text{ minutes.}$$

Otherwise, no spare part j can be provided with 24 hours, and hence it is assumed that

$$\delta_{ijk} = 24 \times 60 = 1440 \text{ minutes.}$$

In the third example where flight k is #556, the same line of reasoning applies, and in this case three different stations can provide a spare part.

Recall that our model is a planning tool to be used at the tactical level. Hence we do not know precisely which station will have a spare part j to provide to station i prior to departure of flight k. Therefore we set δ_{ijk}, the time delay due to failure of part j prior to departure of flight k by the following:

$$\begin{cases} \delta_{ijk} = \min\limits_{i_o \in \Gamma_{ik}} \{\delta_{ijk}(i_o)\} & \mathit{if} \exists\, i_o \\ \delta_{ijk} = 1440 & \mathit{otherwise} \end{cases} \qquad \text{Eq. 2- 2}$$

where $\Gamma_{ik} = \{i_o : i_o$ is the origin station of a flight incoming to i such that its departure time is within 24 hours after the scheduled departure time of k, and $x_{i_o j} > 0\}$.

Let d_{ij} denote the average cost per delay caused by the failure of part j prior to any flight departure at station i. If the airline is a participant to pool part j at station i, then the provider of the part is responsible to provision a replacement to the airline at all time. In this case, the assumption is that no delay occurs. Otherwise, the total cost of delay is a function of the time delay, and it has been shown to be a non-linear function. According to a study done at Air Canada, the cost of delay is assumed to follow the square root function shown in the graph below:

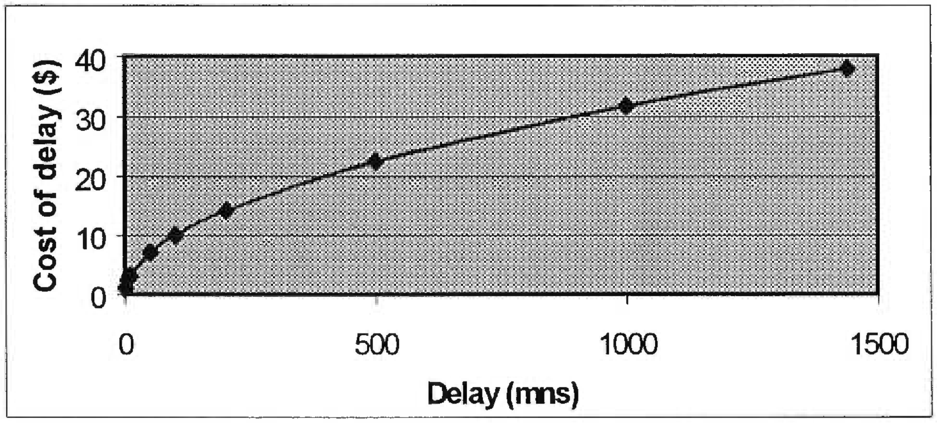

Figure 2: Cost of Delay Function

Let K_{ij} denote the set of flights departing from station i during the week, and that are operated with an aircraft using part j. Hence,

$$\begin{cases} d_{ij} = \dfrac{1}{|K_{ij}|} \sum_{k \in K_{ij}} \sqrt{\delta_{ijk}} & \forall\, i \notin P_{A_j} \\ d_{ij} = 0 & \forall\, i \in P_{A_j} \end{cases} \qquad \text{Eq. 2-3}$$

2.3.2 *Expected removals*

The number of removals per part j is the expected number of times a part fails during a time period. Based on historical data, a Mean Time Between Removals

$(MTBR)$ τ_j, is defined for each part j. τ_j represents the average number of flying hours between two successive removals of part type j.

A part j can belong to one or more types of aircraft. Let $Q(j,u)$ be the quantity of parts j belonging to the aircraft of type u. Let $H(u)$ be the total hours operated by aircraft of type u during the week. Then the total expected number of removals λ_j for part j in a year is calculated as follows:

$$\lambda_j = \sum_u \frac{H(u) * Q(j,u) * 52}{\tau_j} \qquad \text{Eq. 2- 4}$$

In order to calculate the expected number of removals per station, we need to calculate the total number of parts on aircraft of type u used to operate flights leaving from station i. Let $F(i,u)$ be the total frequencies of flights per year with aircraft of type u, departing from station i. Then the expected number of removals λ_{ij} of part j at station i is as follows:

$$\lambda_{ij} = \frac{\lambda_j * \sum_u F(i,u) * Q(j,u)}{\sum_i \left(\sum_u F(i,u) * Q(j,u) \right)}, \qquad \text{Eq. 2- 5}$$

$$\text{and} \quad \lambda_j = \sum_i \lambda_{ij}$$

2.3.3 Process of failure

When failure of parts occurs at a rate less than 10 during a certain period, removals are assumed to follow a Poisson Process. Since λ_j is the expected number of removals of part j during a year, let $k_{\lambda_j t}$ be the Poisson variable representing the number of removals of part j in $[0,t]$. The probability of having *exactly* U *removals* during an interval of time t is given by:

$$\mathrm{P}_p(k_{\lambda_j t} = U) = \frac{e^{-\lambda_j t}(\lambda_j t)^U}{U!} \qquad \text{Eq. 2- 6}$$

The probability of having U *removals or less* during the same time interval t is given by:

$$\mathrm{P}_p(k_{\lambda_j t} \le U) = \sum_{r=0}^{U} \mathrm{P}_p(k_{\lambda_j t} = r) = \sum_{r=0}^{U} \frac{e^{-\lambda_j t}(\lambda_j t)^r}{r!} \qquad \text{Eq. 2- 7}$$

Since λ_{ij} is the expected number of removals of part j at station i during one year, then the probability of having V *removals or less* of part j at station i in $[0,t]$ is:

$$\mathrm{P}_p(k_{\lambda_{ij} t} \le V) = \sum_{r=0}^{V} \frac{e^{-\lambda_{ij} t}(\lambda_{ij} t)^r}{r!} \qquad \text{Eq. 2- 8}$$

When failures of a part j occur at a rate greater than or equal to 10 per period $(\lambda_j \geq 10)$, removals are assumed to follow a Normal distribution, an approximation of the Poisson distribution. Then the number of removals in $[0,t]$ is assumed normal with mean equal to $\lambda_j t$ and standard deviation $\sigma = \sqrt{\lambda_j t}$. Then the probability of having U removals or less during an interval of time t is given by:

$$\mathrm{P}_N(k_{\lambda_j t,\sigma} \leq U) = \mathrm{P}_N\left(z \leq \frac{U - \lambda_j t}{\sigma}\right) \qquad \text{Eq. 2- 9}$$

where $k_{\lambda_j t}$ is a normal random variable with mean $\lambda_j t$ and standard deviation σ equals to $\sqrt{\lambda_j t}$ and z denotes the Standard Normal variable.

2.3.4 *Expected delays*

The variable x_{ij} denotes the initial number of spares of type j allocated at station i. However this number is reduced whenever a spare part is being used. Let $x_{ij}(t)$ be the number of spare parts j available at time of failure t at station i:

$$x_{ij}(t) \leq x_{ij} \quad \forall t \qquad \text{Eq. 2- 10}$$

Two cases have to be considered to calculate the expected delay for station i according to the fact that $x_{ij} = 0$ or $x_{ij} > 0$.

2.3.4.1 $x_{ij} = 0$:

In this case, no spare part is allocated at station i. Hence $x_{ij}(t) = 0$, $\forall t$. Then, when a part j breaks down prior to a flight departure at station i, the time the flight will be delayed is the time it takes to get the earliest supply from a flight coming into station i, originating from station i_o. Station i_o should be able to supply a replacement part from its spare stock (i.e. $x_{i_o j} > 0$). Hence, when $x_{ij} = 0$, the expected number of removals of part j at station i causing a delay is equal to the expected number of removals λ_{ij} (see equation 2-4 and 2-5) of part j at station i.

2.3.4.2 $x_{ij} > 0$:

In this case, a failure of a part j happens at a station i that has an initial allotment of spares. Hence, we need to calculate the probability of having a failure during the time when stock runs out, i.e. when $x_{ij}(t) = 0$ given $x_{ij} > 0$.

As mentioned before, spare inventory is placed at Dorval shop (denoted as station D) to quickly replenish station allotments: when a part at a station becomes unserviceable, it is sent to the Dorval shop for repair. In the case where the shop has no serviceable part to send to the station in replacement of its unserviceable one, the replenishment time R_j, will be the time it takes to repair the station's unserviceable part T_{Aj} plus the transit time T_T. But for the case where Dorval has a serviceable part to send to the station upon receiving the defective one, the replenishment time R_j will be only the transit time T_T. Note that T_T is an average shipping time from and to the station, and is assumed to be independent of part type or location of part. Let I_j be an indicator defined as follows:

$$\begin{cases} I_j = 0 & ,x_{Dj}(t) > 0 \\ I_j = 1 & ,x_{Dj}(t) = 0 \end{cases} \qquad \text{Eq. 2- 11}$$

$$R_j = T_T + I_j T_{Aj}, \quad T_T \text{ is constant } \forall j \qquad \text{Eq. 2- 12}$$

To simplify the computations, the expected number of removals of a part j at the shop D is taken to be equal to the total expected number of removals λ_j ($\lambda_j = \sum_i \lambda_{ij}$). The service level L_j is defined as the probability of replenishment right after a removal of part j is flagged, i.e. when $x_{Dj}(t) > 0$ and therefore, when $I_j = 0$. The probability L_j is computed using the Poisson Distribution function when $\lambda_j < 10$ or using the Normal distribution when $\lambda_j \geq 10$. Given x_{Dj} to be the initial shop's quantity of spares of type j and $x_{Dj}(t)$ the quantity of spares at the time of failure,

$$L_j = \mathrm{P}(x_{Dj}(t) > 0) = \mathrm{P}(R_j = T_T) \qquad \text{Eq. 2- 13}$$

and this probability is approximated by:

$$\mathrm{P}(k_{\lambda_j T_{Aj}} \leq x_{Dj})$$

$(1 - L_j)$ is the probability that Dorval will delay the replenishment until the unserviceable part is repaired, i.e. when $x_{Dj}(t) = 0$ and therefore $I_j = 1$. Then the replenishment time will be $(T_T + T_{Aj})$:

$$1 - L_j = \mathrm{P}(x_{Dj}(t) = 0) = \mathrm{P}(R_j = T_T + T_{Aj}) \qquad \text{Eq. 2- 14}$$

and this probability is approximated by:

$$P(k_{\lambda_j T_{Aj}} > x_{Dj})$$

From the basic definition of expected value or mean λ of a discrete random variable, we know that:

$$\lambda t = 0 * P(k_{\lambda t} = 0) + 1 * P(k_{\lambda t} = 1) + 2 * P(k_{\lambda t} = 2) + \ldots = \sum_{s=0}^{\infty} s\,P(k_{\lambda t} = s)$$

Let μ_{ij} be the expected number of removals of part j at station i per year *causing a delay*. Note that $\mu_{ij} \le \lambda_{ij} \quad \forall i, j$. Let $\mu_{ij}[t]$ be the expected number of removals of part j at station i during a time period t, *causing a delay*. Then given the allotment x_{ij},

$$\mu_{ij}[t] = 0 * P(k_{\lambda_{ij}t} \le x_{ij}) + 1 * P(k_{\lambda_{ij}t} = x_{ij} + 1) + 2 * P(k_{\lambda_{ij}t} = x_{ij} + 2) + \ldots$$

$$= \sum_{s=x_{ij}}^{\infty} (s - x_{ij})\,P(k_{\lambda_{ij}t} = s)$$

$$= \sum_{s=x_{ij}}^{\infty} s\,P(k_{\lambda_{ij}t} = s) - x_{ij} \sum_{s=x_{ij}}^{\infty} P(k_{\lambda_{ij}t} = s)$$

$$= \left(\lambda_{ij}t - \sum_{s=0}^{x_{ij}-1} s\,P(k_{\lambda_{ij}t} = s) \right) - x_{ij}\left(1 - \sum_{s=0}^{x_{ij}-1} P(k_{\lambda_{ij}t} = s) \right)$$

$$= \sum_{s=0}^{x_{ij}-1} (x_{ij} - s)\,P(k_{\lambda_{ij}t} = s) - (x_{ij} - \lambda_{ij}t) \qquad \text{Eq. 2- 15}$$

This derived value is the expected number of removals causing a delay during a period of length t. In this section, we are interested in the expected removals when $x_{ij} > 0$, and during the time when stock runs out at station i, which is the replenishment time R_j. Relying on equations 2-13 and 2-14,

$$\begin{cases} P(R_j = T_T) = L_j \\ P(R_j = T_T + T_{A_j}) = 1 - L_j \end{cases}$$

Therefore, if $x_{ij} > 0$, then the expected number of removals μ_{ij} of parts j causing a delay at the station i is as follows:

$$\mu_{ij} = L_j * \mu_{ij}[T_T] + (1 - L_j)\mu_{ij}[T_T + T_{Aj}] \qquad \text{Eq. 2- 16}$$

Let $\begin{cases} J_{ij} = 0 & x_{ij} > 0 \\ J_{ij} = 1 & x_{ij} = 0 \end{cases}$

be the status indicator of the initial allotment of part j at station i. Recall that the value of d_{ij} given by the equation *2-3* denotes the *average cost of delay per failure* of part j on flights departing from station i when no spare part j is available (when $x_{ij}(t) = 0$). If $D(X_j)$ denotes the expected total cost of delay caused by a failure of parts j given the allotment $X_j = (x_{1j}, x_{2j}, .., x_{sj})$, then:

$$D(X_j) = \sum_{i=1}^{S} d_{ij}\left(J_{ij}\lambda_{ij} + (1 - J_{ij})\mu_{ij}\right) \qquad \text{Eq. 2- 17}$$

It is worthy of note that we have to scan the weekly schedule file in order to evaluate the function $D(X_j)$ for *each* solution X_j. Since the schedule may include up to 3,300 flights, this implies that the function evaluation of $D(X_j)$ is in general costly and computation-intensive to run.

Chapter 3

SOLUTION APPROACH USING GENETIC ALGORITHMS

3.1 Exact Algorithms versus Heuristic Algorithms

Conventional computational techniques, or exact algorithms, are difficult to apply to our optimization problem since, as mentioned before, the objective function is very complex and time consuming to evaluate. Furthermore, it does not include any nice structure to take advantage of to easily evaluate the effect on its value induced by a slight modification of the solution. If the search space is not too large, one can usually develop an enumeration search strategy with appropriate heuristic cutoffs, to keep the computation time under control. If the search space is large, exhaustive search techniques are computationally too expensive and some special artificial intelligence techniques have to be employed. Heuristic or approximate algorithms are often preferred in this case to generate quickly a solution within few percent of the optimum. Genetic algorithms are global optimization techniques that avoid some of the shortcomings exhibited by local search techniques on large search spaces. They are population based heuristic algorithms relying on the genetic inheritance, and allowing a more exhaustive search of solution space. Our approach to solve the optimum distribution of aircraft spare parts uses genetic algorithms as the main search approach.

3.1.1 Exhaustive search

Exhaustive search is the enumeration of all possible solutions in the search space in order to identify the optimal one. Denote by $F(S,N)$ the total number of possible solutions for a part with an allotment of N spares to be allocated at S feasible stations. $F(S,N)$ is formulated as follows:

$$F(S,N)=\binom{N+S-1}{N}=\frac{(N+S-1)!}{N!(S-1)!}.$$

For $N=1$, $F(S,N)=S$.

For $N=2$, $F(S,N)=\dfrac{(S^2+S)}{2}$.

For $S=10$ and $N=1$, the total number of feasible solutions $F(S,N)$ is 10. For $S=10$ and $N=2$, $F(S,N)=55$. Since the total number of feasible stations varies between 10 and 15, the total number of solutions for problems where $N=1$ or 2 varies between 10 and 120 solutions. So for all parts j where $N_j \leq 2$, the exhaustive search method is used in order to find the optimal solution, since the search space of solutions is small. There are 813 part types that have 1 or 2 initial spares.

However, for a part where $S=15$ feasible stations and $N=6$ parts, the total number of feasible solutions increases to *38,760* ($F(S,N)=38,760$). If N incrases to *11*, the search space can be as large as *4,457,400* solutions. The time required to enumerate and evaluate all four million solutions for one part grows rapidly. Therefore, for all part types j where $N_j \geq 3$ (i.e. when the search space is large) a heuristic search using genetic algorithm described in the following section is applied. There are 445 types of parts that have more than 3 initial spares.

3.2 Genetic Algorithms

During the past thirty years, there has been a growing interest in solution procedures based on principles of evolution and inheritance. Genetic Algorithms (GAs) are evolutionary procedures representing a class of adaptive search techniques that have been described and extensively analyzed in the literature (Holland (1975), De Jong (1980), Grefenstette (1986)). John Holland developed the genetic algorithm and provided its theoretical foundation in his book *Adaptation in Natural and Artificial Systems* (1975). A GA is a search procedure modeled on the mechanics of natural selection, rather than of simulated reasoning process. Domain knowledge is embedded in the abstract representation of a candidate solution called Chromosome. Chromosomes are grouped into sets called populations. Successive populations are called generations. A GA is an iterative procedure modifying the current population and maintaining its size constant from one generation to another. At each generation, the chromosomes in the current population are evaluated, and, on the basis of those evaluations, a new population of candidate solutions is generated.

3.2.1 Presentation of the algorithm

An initial population of n chromosomes or solutions is created. All n chromosomes are evaluated, i.e. their fitness is evaluated. At each generation, n chromosomes are chosen for reproduction from the current population. The selection is completed according to the proportional selection. It is a randomized process that ensures that the expected number of times a chromosome is chosen is approximately proportional to that chromosome's performance relative to the rest of the population. Hence the same chromosome can be selected more than once. New chromosomes are generated by means *of genetic recombination operators*. The recombination operator is called *crossover*, and in general, it generates two new solutions called offsprings. Then some perturbations are applied to the offsprings with low frequency via the mutation operator. All the new offsprings are added to the set of parents to create a temporary population of size $2n$. n chromosomes are selected from the temporary population via proportional selection, to create a new generation of size n. This process is repeated until a maximum number of generations is reached. The genetic algorithm used in our study is summarized in Figure 3:

```
    1. Create a temporary initial population of $n$ Chromosomes
       (Generation 0)
    2. Evaluate the fitness of each chromosome
    REPEAT WHILE the number of generations is less than the
    maximum number of generations:
        3. Select $n$ parents from the temporary population via
           proportional selection to create a new generation
        Repeat until all parents are selected ($n$ offsprings are created):
            4. Choose a pair of parents for mating. Apply the
               crossover to create two offsprings
            5. Process each offspring by the mutation operator.
        End Repeat
        6. Evaluate the fitness of each offspring
        7. Add the offsprings to the set of parents to form a new
           temporary population of size $2n$
END REPEAT
```

Figure 3: Simple Genetic Algorithm

Figure 4 below illustrates in a diagram the different steps of the genetic algorithm described in Figure 3:

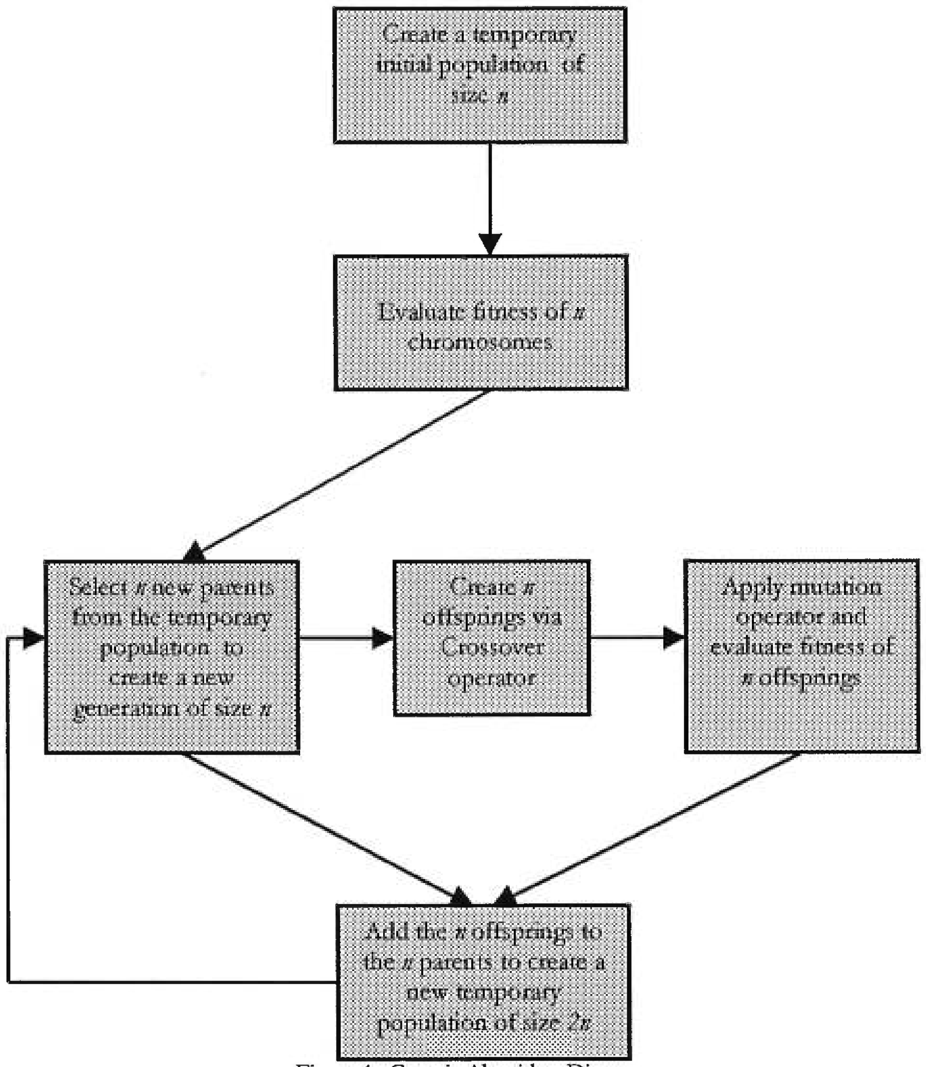
Create a temporary initial population of size *n*
Evaluate fitness of *n* chromosomes
Select *n* new parents from the temporary population to create a new generation of size *n*
Create *n* offsprings via Crossover operator
Apply mutation operator and evaluate fitness of *n* offsprings
Add the *n* offsprings to the *n* parents to create a new temporary population of size 2*n*

Figure 4: Genetic Algorithm Diagram

3.2.2 Encoding of a solution

A crucial operation in using a genetic algorithm to solve a problem is the way of representing or *encoding* a solution. The technique for encoding a solution may vary from one problem to another. In earlier works, researchers used to encode all solutions as bit strings, regardless of the problem. Later, other types of encoding techniques were used. In our case, a bit string encoding is not natural, and we prefer to use the following one:

We refer to the solution vector $X_j = (x_{1j}, x_{2j}, .., x_{sj})$ as the *phenotype* form of the solution. Each component x_{ij} represents the number of parts j allocated at station i. The dimension of the vector solution X_j is S. The solution is encoded into a *genotype* form, representing the artificial chromosome associated with it. The genotype form of a solution is a new vector of dimension N_j, where N_j is the initial number of spare parts j. Each vector component contains the station index where *one* part is allocated. If more than one part is allocated at the station, the station index is repeated for the next vector component. Denote $G_j = (g_{1j}, g_{2j}, ..., g_{N_j j})$ the genotype representation of a vector solution, where $g_{lj} \in A,\ l = 1..N_j$, and A is the set of all stations. For example, for a part j where $N_j = 6$ and $S = 60$, let $X_j = (x_{1j}, x_{2j}, x_{3j}, .., x_{sj})$ $= (1, 0, 2, 0, .., 0, 3)$. Note that $\sum_{i=1}^{S} x_{ij} = N_j = 6$. The genotype vector G_j is represented as $G_j = (1, 3, 3, 60, 60, 60)$.

3.2.3 Population Size

The population size n is the number of solutions or chromosomes per population. The population size remains constant from generation to generation. Population size is a fundamental parameter of any GA. On the one hand, if the selected population size is too small, the algorithm will converge too quickly. On the other hand, a population with too many members offers a larger pool of diverse solutions, but might result in long execution times for significant improvement. Recall that in our case, the evaluation of a solution is very complex and time consuming since the structure of the cost function does not allow to take advantage of the current solution value to obtain the new solution value. Davis in (Davis, 1991), states *that "... the most effective population size is* *dependent on the problem being solved, the representation used, and the operators manipulating the representation"*. For our problem, the population size is proportional to the length of the genotype vector N_j. Experimental results to determine appropriate population sizes are included in chapter 4.

3.2.4 Generation of initial population

The key feature of GAs is their ability to take advantage of the cumulated information about an initially unknown search space in order to bias subsequent search into useful subspaces. Most genetic algorithm implementations do begin with random populations. However, if problem-specific knowledge is available to indicate interesting regions of the feasible domain, then it should be used to guide the process. Initial solutions generated according to problem specific-knowledge are called "seeded" solutions. Only the initial population is seeded with "good" initial members. The need to take advantage of the problem-specific knowledge has been recognized for a long time.

For the Aircraft Spares problem, it seems natural to conjecture that the time delay is smaller for solutions where more parts are assigned to stations with high frequency of departures. To justify this assumption, consider a part j (part *#21-50-155* in our set of data) for which the number of spare parts N_j =*8*, and where the number of feasible stations is *10*. Two different extreme situations are analyzed. In the first situation, ten different solutions are evaluated, where the total number of parts is assigned to *one station*, and none to the other *9* stations. The results are summarized in Table 2. For example, in solution 1, all *8* parts are assigned to the YYZ station and the time delay is equal to *97mns*. In solution 3, all the parts are assigned to the YVR station and the time delay is equal to *187mns*.

Solution	Station	Frequency	Parts @ station	Delay (mns)
#1	YYZ	317	8	97
#2	YUL	134	8	149
#3	YVR	103	8	187
#4	YYC	101	8	178
#5	YHZ	79	8	180
#6	YWG	79	8	196
#7	YOW	53	8	224
#8	YEG	7	8	261
#9	SFO	7	8	246
#10	MIA	1	8	257

Table 2: First Situation: Seeded Solutions

The relation between the frequency of departures at the station where the *8* spare parts are assigned and the time delay of the corresponding solution is illustrated by the graph in Figure 5. This graph indicates that, in general, the time delay decreases when spare parts are assigned to stations with higher frequencies.

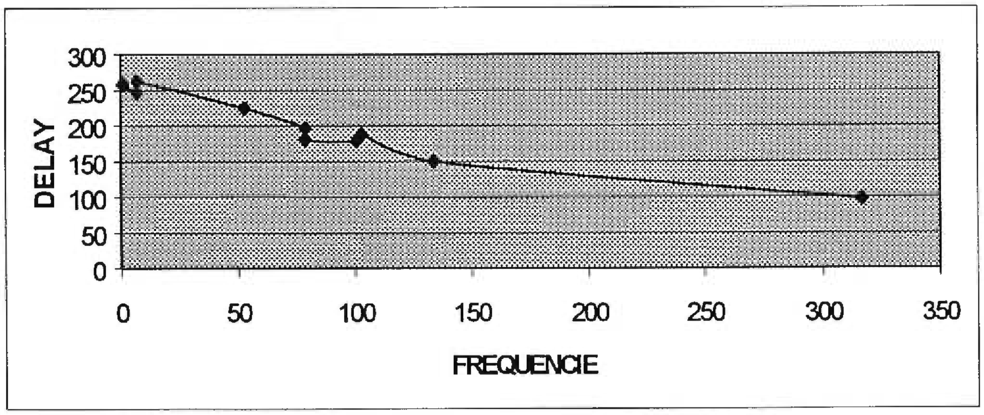

Figure 5: First Situation: Frequency-Delay Relationship

Since the time delay at a station does not depend uniquely on the number of spare parts but also on the providing from the neighboring stations, another situation is analyzed. Here, we use the same part with the same feasible stations, but we assume that *72* spare parts are available. Ten different solutions are evaluated where no spare part is assigned to one station and *8* spares to each of the other *9* stations. The results are summarized in Table *3*, and the relation between the frequency of departures at the station having no spares and time delay of the corresponding solution is illustrated in the graph of Figure 6. For example, in solution 1, no part is assigned to the YYZ station and *8* parts are assigned to each of the other stations, resulting in a time delay of *64mns*.

Solution	Station	Frequency	Parts @ station	Delay (mns)
#1	YYZ	317	0	64
#2	YUL	134	0	40
#3	YVR	103	0	43
#4	YYC	101	0	43
#5	YHZ	79	0	38
#6	YWG	79	0	40
#7	YOW	53	0	40
#8	YEG	7	0	35
#9	SFO	7	0	35
#10	MIA	1	0	34

Table 3: Second situation: Seeded Solutions

The graph in Figure 6 illustrates that, in general, the time delay increases with the frequency of the station having no spare part.

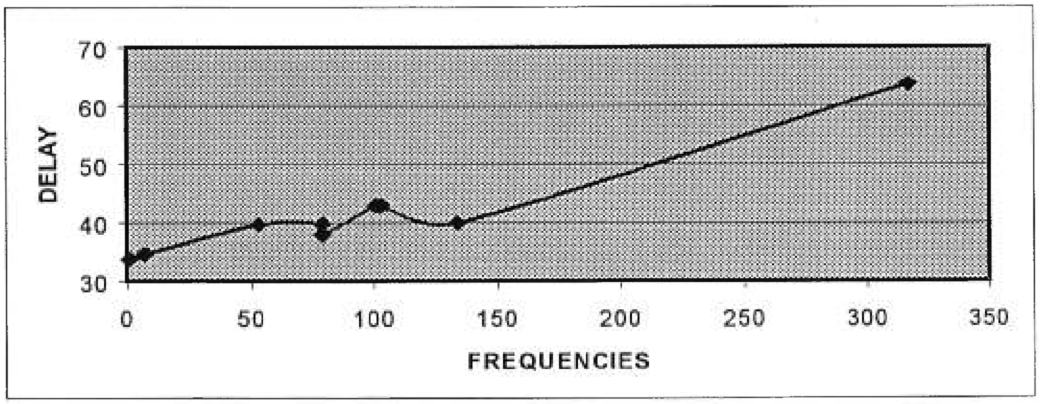

Figure 6: Second situation: Frequency-Delay Relationship

Relying on these observations, the initial population can be partially seeded using frequency of departures per station as a parameter. In this case, some seeded chromosomes are included in the initial population. Each seeded chromosome is specified according to the following proportional selection process. First, the stations are ordered in some sequence. To determine a station in each position of the chromosome, a random positive integer smaller or equal to the total number of flights is generated. Then we select the first station in the sequence such that the sum of its frequency and those of the previous stations in the sequence is greater than or equal to the random number. This proportional selection process ensures that the expected number of times a station is chosen is approximately proportional to that station frequency relative to the rest of the stations. Station indices per generated solution are sorted in ascending order, i.e. for an initial solution $G_j = (g_{1j}, g_{2j}, \dots, g_{N_j j})$, $g_{lj} \leq g_{mj} \quad \forall\, l \leq m$. In chapter 4, numerical results are used to analyze the influence of the percentage of seeded solutions in the initial population and the effect of sorted initial solutions.

3.2.5 Fitness of a solution

The fitness function associates a quality measure to the chromosomes. Selecting solutions based on their fitness value is a major factor in the efficiency of GAs. The greater the fitness value of a chromosome is, the more likely the chromosome is selected for recombination. In a sense, individuals contribute to the gene pool in proportion to their relative fitness. In our problem, the quality of a chromosome is inversely proportional to its cost of delay. Therefore, the fitness of a chromosome solution G_j denoted as $f(G_j)$, is defined to be the inverse of the cost of delay function $D(X_j)$:

$$f(G_j) = \frac{1}{D(X_j)}$$

3.2.6 Proportional Selection & Replacement Strategy

Proportional selection is a randomized selection procedure that ensures that the expected number of times a chromosome is chosen is approximately proportional to its performance relative to the rest of the population. At each generation, n new parents are selected from the temporary population. Each new parent is selected according to a proportional selection process, also known as the roulette wheel parent selection. First the fitness values of all chromosomes in the current population is determined and the chromosomes are ordered in some sequence. Then the following procedure is applied to determine each of the n parents:

1. Generate a random number between 0 and the total of the fitness values
2. Select the first chromosome in the sequence whose fitness value added to the sum of the fitness values of the previous chromosomes in the sequence is greater than or equal to the random number.

As mentioned before, proportional selection process is applied to the temporary population formed by n parents and their n offsprings, in order to form a new generation of n chromosomes (with the exception of the temporary initial solution which contains only n initial solutions). The advantage of this selection technique is that it directly promotes reproduction of the fittest population members. Several other replacement strategies are currently used where only a portion of the population is kept from one generation to another. In chapter 4, numerical results are used to evaluate the efficiencies of other strategies, by keeping a small percentage of best performing chromosomes from generation to generation and selecting the rest of the population via proportional selection.

3.2.7 Crossover Operator

Crossover is an extremely important component of genetic algorithm. It is regarded as the distinguishing feature of GAs and as a critical accelerator of the search process. Crossover is a structured yet randomized information exchange between two chromosomes. The role of the crossover operator is to combine two solutions into an offspring that shares characteristics from both parents. The more widely used operators are the 1-point, the 2-point, and the uniform crossovers. Since the dimension of the vector solution varies between *1* and *11*, only the 1-point and the uniform operators are compared.

In the 1-point crossover, we randomly select a cut point in the parents. Then offsprings are generated by combining the genetic material of one parent on the left-hand side of the cut point with that of the other parent for the positions on the right-hand side (and including the cut point). Figure 7 illustrates an example where two parents are combined via the 1-point crossover.

Parent1	MIA	SFO	YEG	YHZ	YVR	YWG	YUL	YUL	YYZ
Parent 2	SFO	YEG	YHZ	YWG	YWG	YUL	YYZ	YYZ	YYZ
Random Position						↑			
Offspring1	MIA	SFO	YEG	YHZ	YVR	YUL	YYZ	YYZ	YYZ
Offspring 2	SFO	YEG	YHZ	YWG	YWG	YWG	YUL	YUL	YYZ

Figure 7: 1-Point Crossover

In the uniform crossover operator, a random bit string of size N_j is generated. The station in each position of the offsprings is specified according to the bit string as follows: If there is a *0* in a given position of the bit string, the offspring 1 inherits the station of parent 2 for the same position, and offspring 2, that of parent 1. If there is a *1* instead, offspring 1 inherits the station of parent 1, and offspring 2, that of parent 2. Figure 8 illustrates the combination of two parents using a uniform crossover.

Parent1	MIA	SFO	YEG	YHZ	YVR	YWG	YUL	YUL	YYZ
Parent 2	SFO	YEG	YHZ	YWG	YWG	YUL	YYZ	YYZ	YYZ
Random Bit String	*0*	*1*	*1*	*1*	*0*	*1*	*0*	*0*	*0*
Offspring1	SFO	SFO	YEG	YHZ	YWG	YWG	YYZ	YYZ	YYZ
Offspring 2	MIA	YEG	YHZ	YWG	YVR	YUL	YUL	YUL	YYZ

Figure 8: Uniform Crossover

Why is the Crossover operator regarded as a critical feature by the GA practitioners? The point they make is that crossover acts to combine *building blocks* of good solutions from diverse chromosomes.

CHROMOSOME (01101) CONTAINS 32 SCHEMATA:

0##0#   #1###   011#1   #####   01101   0#1#1   ……

Figure 9: Examples of schema in a single bit string

A single chromosome of length 5 in a bit representation is shown in Figure 9, along with some of the building blocks contained in that chromosome. Each building block is represented as a list made up of three characters: 1, 0, and "#". A 1 or a 0 at any position in the building block means that the chromosome must have the same value at that position for it to contain the building block. A "#" at any position in the block means that the value of the chromosome at that position is irrelevant to determining whether the chromosome contains the building block. Holland calls each building block a *schema.* Holland's Schema Theorem says that a schema occurring in chromosomes with above-average evaluations will tend to occur more frequently in the next generation, and one occurring in chromosomes with below-average will tend to occur less frequently. When applying the crossover operator, the best schema should appear more frequently in the population, and their recombination should produce high quality combinations of schemata on single chromosomes.

3.2.8 Mutation Operator

Mutations are perturbations of solutions, occurring with small probabilities. The role of the mutation operator is to randomly perturb solutions in order to maintain an appropriate level of diversity in the population. The mutation operator offers the opportunity for new genetic material to be introduced into a population. The new genetic material does not originate from the parents and it is not introduced into the children via the crossover. Rather, it occurs after crossover is applied, but only a small percentage of the time. As described in step *#5* in Figure 3, each offspring in the population is processed by the mutation operator. With a *1%* probability, every component of each vector solution is replaced by a random feasible station.

3.2.9 Feasibility of generated chromosomes

For some problems, Genetic Algorithms did not provide for successful applications because of their inability to deal with non-trivial constraints. Then, an important issue is to use a proper encoding in order to maintain feasibility of the solutions generated via the crossover and mutation operators.

In our problem, a solution $X_j = (x_{1j}, x_{2j}, .., x_{sj})$ for a part of type j is said to be feasible if it satisfies all four constraints as presented in section 2.2.:

$$\begin{cases} \sum_{i=1}^{S} x_{ij} = N_j \\ x_{ij} \geq 1 \quad \forall\, i \in P_j \\ x_{ij} = 0 \quad \forall\, i \in A_j \\ x_{ij} = 0 \quad \forall\, i \in \overline{M}_j \end{cases}$$

Let the vector $G_j = (g_{1j}, g_{2j}, \ldots, g_{N_j j})$ be the genotype representation of a solution X_j where $g_{lj} \in A,\ \ l = 1..N_j$, and A is the set of all station indices. This representation of a solution facilitates the task of building feasible solutions from one generation to another. In term of the genotype vector G_j, the constraints become:

$$\begin{cases} \|G_j\| = N_j \\ g_{lj} \in P_j \quad \text{for some } l \\ g_{lj} \notin A_j \quad \forall\, l \\ g_{lj} \notin \bar{M}_j \quad \forall\, l \end{cases} \qquad \text{Eq. 2- 18}$$
$$1 \le l \le N_j$$

$\|G_j\|$ is the dimension of the vector G_j.

The following indicates how feasibility is guaranteed for chromosomes generated in the initial population, and for solutions generated when either the crossover or the mutation operators are applied:

Feasibility of 1st constraint: $\|G_j\| = N_j$

All initial solutions G_j include N_j stations. Then, the vector $G_j = (g_{1j}, g_{2j}, \ldots, g_{N_j j})$, for all initial solutions, where $g_{lj} \in A,\ \ l = 1..N_j$ satisfies the first constraint: $\|G_j\| = N_j$. Also, as indicated in section 3.2.7, each offspring generated by the crossover operator has automatically the same vector size as the parent chromosomes. As for the mutation operator, the dimension of a vector solution never changes when the operator is applied. Then, solutions at all generation levels satisfy the first constraint.

Feasibility of 2^nd^ constraint: $g_{lj} \in P_j$ for some l

$P_j \neq \phi$ implies that there exist stations where Air Canada is the provider of the part j to other airlines. In this case, at least one part has to be assigned to each station i belonging to P_j. Then we assign exactly one part to all pooling stations. Let $\|P_j\|$ be the cardinal of the set P_j. Note that it is assumed that $\|P_j\| \leq N_j$, which means that there exist initially enough parts to distribute among the pooling stations. Let $N_j^{'}$ be defined as follows: $N_j^{'} = N_j - \|P_j\|$. The problem is transformed to become of size $N_j^{'}$ with only $N_j^{'}$ parts to be distributed, and $G_j^{'} = (g_{1j}^{'}, g_{2j}^{'}, \ldots, g_{N'j}^{'})$ where $g_{l'j}^{'} \in A,\ l^{'} = 1..N_j^{'}$. Note that the new set P_j for the transformed problem is empty. Then the second constraint is automatically satisfied at all generations.

Feasibility of 3^rd^ and 4th constraints:

Let $C_j = A - (A_j \cup \overline{M}_j)$. Hence C_j is the set of all feasible stations that satisfy the last two constraints. If any initial solution $G_j = (g_{1j}, g_{2j}, \ldots, g_{N_j j})$ is such that each station index is chosen from the set C_j, then it verifies the 3rd and 4th constraints. Now, since any offspring generated by the crossover operator inherits a combination of station indices from the parents, it also satisfies these constraints. Finally, feasibility is maintained during mutation if a station index g_{lj} is replaced by an index from the set $C_j = A - (A_j \cup \overline{M}_j)$.

Thus, feasibility is easily maintained without any additional computational effort. This is one of the reasons that make Genetic Algorithms attractive to solve the Spare Parts problem.

Chapter 4

NUMERICAL RESULTS

Choosing a suitable population size, the right combination operator or even a reasonable stopping criteria are key decisions faced by all genetic algorithm users. Parameter settings greatly influence performance. Choosing the best values for these parameters is generally difficult, mainly because of the interactions existing between parameters, and also, because of the long time required to evaluate a single set of parameters. In the following sections, we consider different values for each of the GA parameters mentioned in the previous chapter. Some of the parameters are tested individually and some dependent parameters are tested when combined. In order to test one parameter, all the others are fixed and the tested parameter is assigned various values. The following specifies the part types used in the tests and the selection criteria for the best parameter values.

In order to make firm conclusions on the performance of each parameter value, there should be a good sample of part types, each having different initial number of spares. Three part types are chosen and tested throughout most of the experiments. As shown in Table 4, one part type has 3 initial spares, the second has 6 spares and the third has 9 spares. Note that initial quantities of spares for all part types where GA is applied vary between 3 and 11 spares. Each part type belongs to a different aircraft type, but each fleet covers a similar network of stations. The fourth column in Table 4 shows the number of parts per aircraft.

The number of flight departures per week is also given for each fleet type. The last column shows the number of feasible stations covered by the fleet type.

PART # / TYPE j	INITIAL QUANTITY N_j	AIRCRAFT TYPE(S)	QUANTITY PER AIRCRAFT	DEPARTURES PER WEEK	FEASIBLE STATIONS
21-10-157	3	DC9-15 DC9-32	2 2	864	10
21-50-373	6	B767-233	9	410	10
21-20-365	9	A320	2	1222	12

Table 4: Part types

For each parameter testing, 10 tests are performed for each of the three part types. The "Best Value" column shown in the test tables below is the minimum delay cost achieved among the total number of runs, and which "might" be the optimal solution. The average cost of delay for the 10 tests is calculated along with the confidence interval, with 95% confidence level. A confidence interval of $\pm\beta$ indicates that for 100 tests, 95 out them have a cost of delay belonging to the interval $D(X_j) \pm \beta$, where $D(X_j)$ is the mean of the 100 tests. The number of tests resulting in the best value out of the 10 tests is also given. Finally, the last two columns of the show the average number of solutions explored or generated *per test*, along with the average time of execution. The computer used for the tests is PC based Toshiba 4030CDS, 233 MHZ.

4.1 Stopping criteria

Several stopping criteria exist for the genetic algorithm. The algorithm may be stopped when all chromosomes in the generation are identical, i.e.

$f(G_j^m) = f(G_j^n)$ for all G_j^m and G_j^n solutions belonging to the same generation. However, a reasonable number of solutions must be explored. A convergence criterion may not be efficient, especially if it generates as many solutions as the exhaustive search. Such a criteria is too costly knowing that the evaluation of the cost of delay $D(X_j)$ per chromosome or per solution X_j is long (see section 2.3 Evaluation of a solution). The alternative criterion used in the tests below is to stop the process after a fixed number of generations. This fixed number of generations was set to be $10 * \lfloor N_j/2 \rfloor$, where N_j is the problem size or the initial number of spares for part j. This stopping criterion results in exploring a reasonable number of solutions and obviously not exceeding the total number of feasible solutions. This choice seems appropriate as confirmed in the following sections when testing different parameters.

4.2 Tests of population size

The population size n is the number of solutions or chromosomes generated per population. The population size remains constant from generation to generation. Population sizes are usually a function of the problem size and are directly proportional to the length of the genotype vector N_j: $n = \alpha N_j$, where α is a positive integer. Three different values of α are tested for each of the three part types. The results are summarized in Table 5. The parameters are fixed as follows:

- Population Size test: $n = 3N_j$ <u>or</u> $n = 5N_j$ <u>or</u> $n = 7N_j$ ($\alpha = 3, 5$ *or* 7).
    - Mutation operator rate: 1%
    - Percentage of seeded solutions in initial population: 50%
    - Percentage of best solutions for new generation: 10%
    - Crossover Operator: Uniform
    - Initial solutions are sorted: $G_j = (g_{1j}, g_{2j}, \ldots, g_{N_j j})$, $g_{lj} \leq g_{mj} \quad \forall\, l \leq m$.

Problem Size Nj	Best Value	Population Size *n*	Avg. Cost of Delay D(Xj)	Confidence Interval	Total Best Solutions	Solutions generated	Time (mns)
3	17.80	*3Nj*	18.33	±0.59	6	90	2.2
		5Nj	17.82	±0.02	8	150	3.6
		7Nj	18.02	±0.27	8	210	4.9
6	7.20	*3Nj*	7.29	±0.06	5	539	7.2
		5Nj	7.24	±0.04	7	900	11.8
		7Nj	7.21	±0.02	9	1260	16.3
9	91.00	*3Nj*	92.41	±1.48	5	1079	19.0
		5Nj	91.27	±0.50	9	1800	30.4
		7Nj	91.00	±0.00	10	2520	42.8

Table 5: Population Size Test Results

For $\alpha = 3$, 5 to 6 runs out of the 10 tests resulted in the best value for each of the 3 types. As for $\alpha = 7$, the results are much better. For example, for the tests of the third part type, 10 tests out of 10 resulted in the best value. As might be expected, larger populations lead to better performance because of the larger pool of diverse schemata available in a larger population. However, the bigger the population size is, the more solutions are evaluated, and therefore solution time increases. The results when $\alpha = 5$ are very close to the ones when $\alpha = 7$, and run times are more manageable. Also, the confidence intervals for the 10 tests are very short. Therefore, the population size $n = 5N_j$ is selected for the rest of the numerical results.

4.3 Tests of seeded initial population

As it has been indicated in Section 3.2.4, the delay is inversely proportional to the frequency of departures. Therefore, solutions that are seeded according to the selection criteria in Section 3.2.4 perform better than randomly generated solutions. However, the presence of too many highly fit chromosomes in the initial population results in premature termination at a local optimum. In order to identify the best strategy, three values are tested for the percentages of seeded initial solutions:

- Percentage of seeded solutions in initial population: 20% <u>or</u> 50% <u>or</u> 80%
    - Population Size $n = 5N_j$
    - Mutation operator rate: 1%
    - Percentage of best solutions for new generation: 10%
    - Crossover Operator: Uniform
    - Initial solutions are sorted: $G_j = (g_{1j}, g_{2j}, \ldots, g_{N_j j})$, $g_{lj} \leq g_{mj} \quad \forall\, l \leq m$.

Problem Size Nj	Best Value	Seed Percentage	Avg. Cost of Delay D(Xj)	Confidence Interval	Total Best Solution	Solutions Generated	Time (mns)
3	17.80	20%	18.29	±0.50	5	176	5.2
		50%	17.80	±0.00	10	150	4.3
		80%	18.23	±0.41	7	160	4.7
6	7.20	20%	7.25	±0.05	7	930	12.6
		50%	7.24	±0.04	7	930	12.1
		80%	7.26	±0.06	7	900	11.7
9	91.00	20%	91.47	±0.48	7	1804	30.00
		50%	91.27	±0.50	9	1804	30.50
		80%	91.24	±0.30	8	1804	30.10

Table 6: Initial Population Test Results

An initial population with 50% seeded solutions seems to generate better results than with 20% or 80%. Figure 10 illustrates how the cost of delay decreases during the resolution for each percentage (20%, 50%, 80%) for the problem with N_j=3.

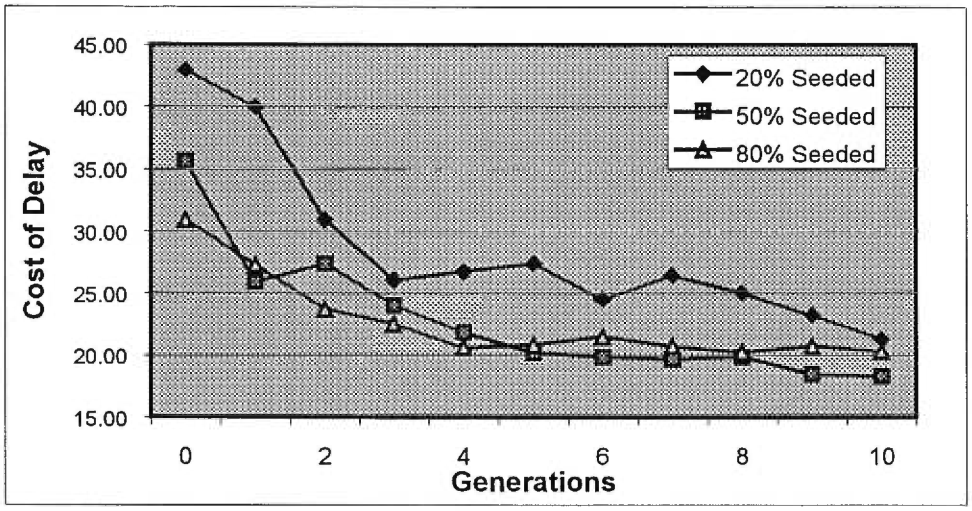

Figure 10: Seeded Population Comparison

Each point on the graph represents the average cost of delay for the n solutions. As expected, for initial population with 20% seeded solutions (i.e. 80% random solutions), the average cost is initially higher, and it takes longer to converge to a minimum. However, for initial population with 80% seeded solutions, the average cost is initially better and it converges rapidly, reaching sometimes a local minimum. With a 50% seeded initial population, the algorithm seems to converge to optimal solutions more likely than with 20% or 80%.

4.4 Tests of offsprings selection

A new generation can be created by completely replacing the parent generation by a new one obtained via proportional selection, or by keeping a small percentage of best performing chromosomes and selecting the remainder of the population via proportional selection. The results for four different strategies are summarized in Table 7:

- Percentage of best solutions: 0% or 4% or 10% or 25%
  - Population Size $n = 5N_j$

- Mutation operator rate: 1%
- Percentage of seeded solutions in initial population: 50%
- Crossover Operator: Uniform
- Initial solutions are sorted: $G_j = (g_{1j}, g_{2j}, \ldots, g_{N_j j})$, $g_{lj} \le g_{mj} \quad \forall\, l \le m$.

Problem Size Nj	Best Value	% Best Solutions	Avg. Cost of Delay D(Xj)	Confid. Interval	Total Best Solutions	Solutions Generated	Time (mns)
3	17.80	0%	18.23	±0.30	4	176	4.00
		4%	17.91	±0.18	8	176	4.60
		10%	17.80	±0.00	10	176	4.25
		25%	18.01	±0.40	7	176	4.30
6	7.20	0%	7.59	±0.13	2	930	12.70
		4%	7.28	±0.05	4	930	12.20
		10%	7.24	±0.04	7	930	12.13
		25%	7.22	±0.02	8	918	12.00
9	91.00	0%	94.71	±2.66	4	1804	31.02
		4%	92.17	±0.99	6	1804	30.00
		10%	91.27	±0.50	9	1804	30.50
		25%	91.00	±0.00	10	1804	31.50

Table 7: Offsprings Selection Test Results

When the best performing solutions are not forced to remain from generation to generation (best solutions percentage = 0%), they tend to be lost. Hence, the results in Table 7 indicate that the algorithm takes longer to regenerate good performing solutions. However, when the percentage selection is 10% or 25%, the results for all 3 problem-tests are very good. For the problem when N_j=3, a percentage selection of 25% seems to be a bit too aggressive as 3 out of 10 tests converges to a local minimum.

4.5 Tests of crossover operator

The role of the crossover operator is to combine two solutions into an offspring that shares characteristics from both parents, by exchanging parts of their corresponding chromosomes. The two operators that are tested in this section are the uniform crossover and the 1-point crossover operators. An additional component is added to the tests of the crossover operators in order to analyze the relevancy of sorting the chromosomes in the initial solutions. Both parameters are tested jointly since the value of one can be dependent on the other. The results summarized in Table 8 are obtained using the following parameters:

- Crossover operators: Uniform or 1-point.
- Initial solutions are sorted: $G_j = (g_{1j}, g_{2j}, \ldots, g_{N_j j})$, $g_{lj} \leq g_{mj} \quad \forall\, l \leq m$ or non-sorted.
    - Percentage of best solutions: 10%
    - Population Size $n = 5N_j$
    - Mutation operator rate: 1%
    - Percentage of seeded solutions in initial population: 50%

The results for the problem where $N_j = 6$ are not very conclusive, since the average values for the 4 combinations of the two parameters are very similar and the confidence intervals are small. Now, for the problems with $N_j = 3$ and $N_j = 9$, the results in Table 8 show that sorting the initial solutions (i.e. sorting the station indices within each chromosome), seems to allow generating better chromosomes, and therefore the algorithm converges to better solutions. The behavior of the algorithm with sorted initial solutions can be somehow explained by Holland's Schema Theorem described in section 3.2.7. When station indices in a chromosome are sorted, each position in the schema will have a smaller number of values to be assigned to, compared to when solutions are not sorted. By sorting the solutions, the schema theory can be applied and is more likely preserved than having each station index assigned to a random position.

Problem Size Nj	Best Value	Crossover Operator	Sorted initial solutions	Avg. Cost of Delay D(Xj)	Confid. Interval	Total Best Solution	Time (mns)
3	17.80	Uniform	No	18.06	±0.26	4	4.50
		1-Point	No	17.92	±0.33	8	4.10
		Uniform	Yes	17.80	±0.00	10	4.25
		1-Point	Yes	17.93	±0.33	7	4.60
6	7.20	Uniform	No	7.25	±0.10	8	11.60
		1-Point	No	7.22	±0.04	8	11.60
		Uniform	Yes	7.24	±0.07	7	12.13
		1-Point	Yes	7.27	±0.10	6	11.60
9	91.00	Uniform	No	94.71	±0.77	7	30.30
		1-Point	No	92.02	±1.28	5	30.90
		Uniform	Yes	91.27	±0.50	9	30.50
		1-Point	Yes	91.36	±0.55	7	31.50

Table 8: Crossover Test Results

The uniform crossover is more efficient than the 1-point crossover in 5 out of 6 cases, as shown in the "Total Best Solutions" column. This result is consistent with the one obtained in Syswerda, 1989 (Section 4.6 - Other problems of various sizes).

Other results are obtained for other problems including different number of spare parts ($3 \leq N_j \leq 11$), and belonging to different aircraft types. Table 9 includes the average results taken over 10 tests for each problem. The percentage variance of the average cost from the best value (shown in the 4th column) varies between 0.0% and 1.3%. These results are very encouraging, showing the good performance of the Genetic Algorithms to solve the Spares problem.

Problem Size Nj	Best Value	Avg. Cost of Delay D(Xj)	% Variance From Best Value	Total Best Solutions out of 10 tests
3	17.8	17.8	0.0%	10
4	11.8	11.93	1.1%	9
5	26.20	26.28	0.3%	6
6	7.20	7.24	0.6%	7
7	88.80	88.99	0.2%	9
8	34.70	35.15	1.3%	7
9	91.00	91.27	0.3%	9
10	59.1	59.1	0.0%	10
11	62.40	62.48	0.1%	7

Table 9: Test Results for various part types

Chapter 5

A COMPARISON WITH THE METRIC MODEL AND OTHER RESEARCH

5.1 METRIC Model Description

During the past four decades, a substantial number of mathematical models have been developed relating to the management of repairable inventory systems. A well-known mathematical model called METRIC, or Multi-Echelon Technique for Recoverable Item Control, was formulated over 30 years ago by C.C. Sherbrooke (1968). This optimization problem is capable of determining base and depot stock levels (two-echelon inventory system) for recoverable items – items subject to repair when they fail. METRIC was specially designed for the Air Force and The RAND Corporation (see W. S. Demmy (1981)). METRIC is designed for application at the weapon-system level, where a particular item may be demanded at several bases or stations, and the stations are supported by one central depot. The item demand for the METRIC model follows a compound Poisson distribution with a mean estimated by a Bayesian procedure. When a failure occurs at a base, a demand is placed on base supply to replace a corresponding part. Depending on the nature of the failure, the failed part is then either repaired at that base with a probability r, or is sent to the depot for repair with a probability $(1-r)$. If repair is done at the base, re-supply of base supply comes from the base maintenance, otherwise, the re-supply comes from the depot. Thus, the inventory policy for placing orders on the base's maintenance organization or the depot is an $(s-1,s)$ policy. It is a policy that replenishes back to level s once the inventory level drops to $(s-1)$, which means that items are not batched for repair or re-supply requests. The $(s-1, s)$ is an appropriate policy for high-cost, low-demand items. No lateral re-supply between bases is considered in the METRIC model.

5.2 Model Purposes

As stated in "METRIC: Multi-Echelon Technique for Recoverable Item Control" by C.C. Sherbrooke (1968), METRIC has three main purposes:

1. *Optimization:* A governing purpose of the model is to determine optimal base and depot stock levels for each item, subject to a constraint on system investment or system performance.
2. *Redistribution:* The model takes fixed stock levels for each item, and optimally allocates or redistributes the stock between the bases and the depot.
3. *Evaluation:* The model evaluates the performance and investment cost for any allocation of stock between the bases and the depot.

Optimization is of prime interest in the early acquisition phase. By considering the unit cost of the item, the METRIC approach focuses management attention on the entire weapon system, so that an appropriate combination of system effectiveness and system cost can be selected. Optimization targets are expressed as system investment in dollars or expected backorders per item. *Redistribution* is a major concern when items are in long or short supply. For example, when no new procurement budget is available, the decision to redistribute might depend on the increased effectiveness over the current distribution levels. When budget is available, redistribution decision is based on comparing the cost of the redistribution with the cost and effectiveness of a new procurement. *Evaluation* is important throughout the life of a weapon system.

5.3 A comparison with the Allotment of Aircraft Spare Parts model

The fact that our problem aims at allocating aircraft parts given their initial quantities, and not at determining their quantities based on investments or effectiveness constraints, we are then only interested in comparing it with the METRIC concepts of redistribution. In his article, Demmy (1979) shows the major discrepancies between an actual Air Force redistribution problem and METRIC assumptions. Some differences and also some similarities exist between the METRIC algorithm and our allocation problem. This is also due to the fact that our allocation assumptions are different than the Air Force redistribution assumptions. Some of the major differences are:

1. No lateral re-supply between bases or stations is considered in the METRIC model. However, the fact that parts are transported on one of the Air Canada scheduled flights, receiving a serviceable part from a neighboring station onboard a flight is the most efficient and cost effective way to re-supply.
2. In the METRIC model, demand obeys a compound Poisson process. It is a logarithmic Poisson process obtained by considering batches of demand where the number of batches follows a Poisson process and the number of demands per batch has a logarithmic distribution. In our model, demand or failure does not happen by batches. It is assumed to follow the Poisson distribution or Normal distribution when failure rate is high.
3. The objective of the METRIC model is to minimize backorders. A backorder exists when there is an unsatisfied demand at the base and is dependant on how quickly the base repair shop or the depot repair shop can provide a serviceable part. Our objective in the aircraft allotment model is to minimize the cost of flight delays due to unsatisfied demand. However, the length of flight delays is

dependent on the stock levels of other stations or bases and also on the repair shop's stock level.

4. With a probability r, repair is done at the base level for the METRIC model, otherwise, it is sent to the depot for repair. As for the Allotment model, all repairs are made at one central repair shop which is the Dorval shop.

Some of the similarities of both models are:

1. The demand process is stationary over the prediction period, and the parameters of the demand process are known. Even though those parameters are dependent on the aircraft flying hours and therefore change with the revision of the schedule, the assumption is still valid due to the fact that redistribution decisions are made every time the schedule changes.
2. Both models assume infinite repairs with no condemnations. This assumption is said to be conservative if redistribution does not happen frequently. However, the fact that aircraft parts have low condemnation rates, initial quantities of spares are adjusted with every redistribution of stock, to most effectively meet current conditions.
3. Both models assume infinite repair capabilities at the depot. As Albright (1989) has pointed out, models assuming infinite repair capacity always underestimate the amount of congestion in the system. More recently, Kim *et al.*(2000) studied the model under finite repair capacities.

5.4 Other research

Since the publication of the METRIC model by Sherbrooke (1967), substantial work has been devoted to extend and refine the METRIC model (see Cho and Parlar (1991)). Fox and Landi (1970) suggested a Lagrangian approach for solving the metric problem. Later on, Muckstadt (1978) developed a method for estimating the value of the Lagrangian multiplier, reducing computation times, and also suggests a simple approximation for estimating the optimal depot stock level. Later on, Muckstadt and Thomas (1980) show that adaptations of single location methods are inferior to method designed to take advantage of a system's structure, a multi-echelon inventory system. Graves (1985) considered a two-echelon inventory system with a small variant to Sherbrooke's METRIC model. The main difference is that Graves does not allow repairs at the bases. On a set of test problems, he showed that his model provides a more accurate approximation than the METRIC model. More recently, Zorn *et al.*(1999) introduced a piece-wise linear modeling framework for the United States Air Force hierarchical inventory model.

Muckstadt (1973) also proposed a model for multi-indenture parts. This model, called MOD-METRIC, an extension of METRIC, permits the explicit consideration of a hierarchical parts structure. The question of reparable spare parts is an issue in both the public and private sectors. Tedone (1989) describes a system utilized by American Airlines, the Rotables Allocation and Planning System, or RAPS, that has similarities to the METRIC system. It is a PC-based decision support system that provides forecasts for the future demand of rotable parts as a first phase, and recommends the allocation of parts among the stations with least-cost for the second phase. To calculate total expected system demand per part type, RAPS uses linear regression to establish a relationship between part removals and flying hours. In the second phase, the demand forecast or the initial quantity of parts required for the system is distributed among the stations according to weights assigned to each station. Those weights reflect the level of activity evaluated through the flight schedule, the history of part removals and the maintenance schedules.

CONCLUSION

Since the problem includes a small number of constraints, Genetic Algorithms seem to be very appropriate to deal with the Spare Parts problem. The results indicate that on average, the solutions generated are within 1.3% of the best solution or optimum solution. This is the case for all problem sizes, for various part types and aircraft types.

It is worth to note that the algorithm is computation intensive. Originally, the project was designed for the operation of Air Canada requiring to run the optimization algorithm once a month in order to redistribute the parts when the flight schedule changes. The evaluation of a solution is making the algorithm computationally expensive, as the delay is estimated for every single flight of the week. Should this optimization be required on a more frequent basis, or within few hours, the cost of delay could be simulated instead of being fully evaluated. By doing so, the execution time is reduced, and the reallocation of the parts can be done more frequently.

Another avenue for future research would be to identify a more specific crossover operator, by taking advantage of the problem structure.

BIBLIOGRAPHY

Albright S.C., *An approximation to the Stationary Distribution of a Multiechelon RepairableIitem Inventory System with Finite Sources and Repair Channels.* Nav Res Logist Q36: 179-195 (1989)

Buckles B. and Petry F., *Genetic Algorithms*, IEEE Computer Society Press, Los Alamitos, California (1992)

Cho D.I., Parlar M., *A Survey of Maintenance Models for Multi-Unit Systems*, European Journal of Operational Research 51: 1-23 (1991)

Davis L., *Genetic Algorithms and Simulated Annealing*, Pitman, London, England (1987)

Davis L., *Handbook of Genetic Algorithms*, ed., Van Nostrand Reinhold, New York, New York (1991)

Demmy W.S., *On the METRIC Distribution Algorithm*, Modeling and Simulation 10: 523-588 (1979)

Demmy W.S., Presutti V.J., *Multi-Echelon Inventory Theory in the Air Force Logistics Command*, TIMS Studies in the Management Sciences 16: 279-297 (1981)

Fox, Bennett, and Landi M., *Searching for the Multiplier in One-Constraint Optimization Problems*, Operations Research 18: 253-262 (1970)

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

Graves S.C., *A Multi-Echelon Inventory Model for a Reparable Item for one Replinishment*, Management Science 31, 1247-1256 (1985)

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

Kim J.S., Shin K.C., Park S.K., *An Optimal Algorithm for Repairable Item Inventory System with Depot Spares*, Journal of the Operational Research Society 51: 350-357 (2000)

Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag (1992)

Muckstadt J.A., *A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System.* Management Science 20: 472-481 (1973)

Muckstadt J.A.., *Some Approximation in Multi-Item, Multi-Echelon Inventory Systems for Recoverable Items.* Nav Res Logist Q25: 377-394 (1978)

Muckstadt J.A. and Thomas L.J., *Are Multi-Echelon Inventory Methods Worth Implementing in Systems with Low Demand Rate Items?* Management Science 26: 483-494 (1980)

Sherbrooke C.C., *METRIC: Multi-Echelon Technique for Recoverable Item Control*, Operations Research 16: 122-141 (1967)

Syswerda G., *Uniform Crossover in Genetic Algorithms,* Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, California, Morgan Kaufmann Publisher: 2-9 (1989)

Tedone M.J., *Reparable Part Management*, Interfaces 19:4: 61-68 (1989)

Zorn W.L., Deckro R.F., Lehmkuhl L.J., *Modeling Diminishing Marginal Returns in a Hierarchical Inventory System of Repairable Spare Parts*, Annals of Operations Research 91: 319-337 (1999)

ACKNOWLEDGMENTS

I wish to express sincere appreciation to Professor Jacques Ferland, my thesis director, for having helped me in realizing this project. His dedication to my work and to his students has been very apparent. I thank him for his unwavering patience, and valuable advice.

Special thanks to Professor Robert Cléroux, my thesis co-director, for his assistance in the preparation of this thesis. I thank him for the many drafts that he has patiently reviewed, and for his observant council.

Thank you to Air Canada and to my former boss Yvan Corriveau, for giving me the opportunity to work on such an interesting operational project. My appreciation and love to my colleagues Soad, Nojoua, and Michel Courval.

My deep thanks go to my family and my cousin Micha for their support and encouragement. Thank you Micha for your inspirational words. My gratitude and love to my sister Grace. Thank you for your faith in my abilities, for always having something encouraging to say. My special thanks and appreciation to my friend Andrew. Thank you for your continuous support throughout the year, for your elevating and encouraging words. Thank you for being there.