Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

# Comparison and Evaluation of Multi-Objective Genetic Algorithms for Military Planning and Scheduling Problems

*Applied to Course of Action Planning*

*Adel Guitouni*
*DRDC Valcartier*

*Lamia Belfares*
*Université Laval*

**Defence R&D Canada – Valcartier**
Technical Report
DRDC Valcartier TR 2003-372
April 2008

Canada

# Comparison and Evaluation of Multi-Objective Genetic Algorithms for Military Planning and Scheduling Problems

*Applied to Course of Action Planning*

Adel Guitouni
DRDC Valcartier

Lamia Belfares
Université Laval

**Defence R&D Canada – Valcartier**

Technical Report
DRDC Valcartier TR 2003-372
April 2008

Principal Author

Adel Guitouni
Defence Scientist

Approved by

Eloi Bossé
S/H C2 Decision Support System Section

Approved for release by

Christian Carrier
Chief Scientist DRDC Valcartier

# Abstract

Planning detailed military Courses of Action (COAs) is a very complex and time consuming. In this work, we investigate Evolutionary algorithms (EAs) to solve COAs'resource management and scheduling problems. The performance of such algorithms is assessed based on their outcomes quality. Efficient algorithms exhibit a good approximation of the Pareto optimal sets while requiring reasonable computational resources. A good approximation set is the resultant of a trade-off between diversity of solutions and their proximity to the true Pareto front. Such a balance is difficult to achieve with NP-hard problems exhibiting Pareto frontier discontinuity and multimodality although EAs are able to handle such optimization features. In this report, the multicriteria filtering genetic algorithm (MFGA) is proposed to achieve balanced proximity-diversity of the generated solutions. It uses a reproduction procedure based on a multicriteria filtering method and dominance concept to select solutions characterized by at least one best-scored objective or by all objectives achieving minimal threshold values. It is applied together with crossover and mutation operators. We illustrate this multi-objective EA in an enlarged sampling size scheme to solve highly constrained course of action planning problems. Cardinal and ordinal objectives are considered. An empirical comparison with three state-of-the-art multi-objective EAs is done using metrics of performance. The results show that this new approach is highly competitive, especially when applied to high dimensional problems such as those frequently encountered in real world applications.



# Résumé

La planification de suites d'actions militaires (COAs) est très complexe et demande beaucoup de temps. Dans ce travail, nous examinons des algorithmes évolutionnaires (AE) pour résoudre la gestion de ressources et des problèmes d'ordonnancement d'une COA. Les AEs sont de plus en plus utilisés pour résoudre des problèmes d'optimization. La performance de tels algorithmes est évaluée sur la base la qualité des solutions générées. Les algorithmes sont efficaces s'ils sont capables de trouver de bonnes approximations des solutions Pareto optimales tout en exigeant des ressources informatiques raisonnables. Un bon ensemble de solutions est celui qui permet de trouver un compromis entre la diversité des solutions et leur proximité de la frontière Pareto. Un tel équilibre est difficile à réaliser avec des problèmes NP-Durs. Dans le cadre de ce rapport, nous proposons un algorithme génétique basé sur le filtrage multicritère (MFGA) pour générer des solutions de cette qualité. MFGA utilise une procédure de reproduction basée sur une méthode de filtrage multicritère et le concept de dominance pour choisir des solutions caractérisées par la meilleure évaluation sur un des objectifs ou des évaluations supérieures à un certain seuil pour tous les objectifs. Des objectifs cardinaux et ordinaux peuvent être considérés. Cette procédure est appliquée séquentiellement aux opérateurs de croisement et de mutation. Une comparaison empirique de l'algorithme proposé avec trois AEs de la littérature est faite en se basant sur des métriques de performance. Les résultats montrent que cette nouvelle approche est très compétitive particulièrement pour des problèmes de grandes tailles, ce qui est le cas des problèmes de planification militaire.

This page intentionally left blank.

# Executive summary

## Comparison and Evaluation of Multi-Objective Genetic Algorithms for Military Planning and Scheduling Problems: Applied to Course of Action Planning

**Guitouni, Adel; Belfares, Lamia; DRDC Valcartier TR 2003-372; Defence R&D Canada – Valcartier; April 2008.**

Course of Action (COA) Development is a very important enterprise. The Commander's guidance and intent should be understood and translated into comprehensive and flexible plans within the time available. COAs "should answer the fundamental questions of when, who, what, where, why and how". Each COA should be suitable, feasible, acceptable, exclusive and complete. A good COA positions the force for future planned operations and provides flexibility to meet unforeseen events during its execution. The "who" in a COA does not specify individual units, but rather uses generic assets and capabilities. During the development phase, staffs analyse the relative combat power of friendly and enemy forces, and generate comprehensive COAs.

The challenge for the planning officers is to generate complex, spatially and temporally interdependent activities with precedence relationships, subject to resource constraints, while satisfying multiple incommensurable and often conflicting criteria. We propose modelling a COA planning as a multiple mode resource-constrained project-scheduling problem (MRCPS). From a methodological point of view, planning and scheduling are quite similar. The model decomposes generic activities (tasks with specific combination of capabilities) into elementary (or simple) inter-related actions to accomplish the objectives of the mission. We assume that a complex mission is decomposable into granular simple activities with appropriate dependencies. This process implies the identification of the tasks (what) as well as their dependency relationships (when and where), the pool of available capabilities (how) with their location, and finally the objectives of the mission (why). A COA is then represented as an oriented time-space graph. Depending on the combination of resources allocated, the action's position in the schedule, and uncontrolled events (enemy and civilian activities, weather, etc.), different COA networks could be obtained. They constitute variants (or alternatives) of how the mission can be achieved with different evaluations on objectives and predictions for uncontrolled event occurrences.

Solving COA planning problems is NP-Hard. This means that it is impossible to solve such problem with exact methods. To obtain promising feasible alternatives with respect to multiple objectives, we thought to explore the potential of evolutionary algorithms (EAs). EAs are meta-heuristics that already have been proven to be reliable for solving combinatorial NP-hard problems.

Several approaches have been proposed to deal with the multi-objective aspect of resource-constrained project-scheduling (RCPS) problems. The most common one aggregates all objectives into one single scalar value by using weighted sum, or similar functions, according to the decision maker's preferences set. The search is then performed several times to find a compromise solution that reflects these preferences. Another approach is to generate the set of compromise or non dominated solutions. This has increased the interest for the Pareto

optimization and multi-objective EAs. However, applications to scheduling problems are still scarce (Landa Silva and Burke, 2002).

In this work, different COA variants are generated using a specific evolutionary algorithm: the genetic algorithms (GAs). GAs are able to deal simultaneously with multiple solutions for solving multi-objective optimization problems. This allows finding a set of potential Pareto optimal solutions in the same iteration without any sensitivity to the shape or the continuity of the Pareto front (Coello Coello, 1999). The set of potential Pareto solutions is obtained here by investigating different procedures based on the dominance concept combined with a new approach referred to as the 1st order multi-criteria filtering method. This work is a part of project 13dm (mission planning systems).

# Sommaire

## Comparison and Evaluation of Multi-Objective Genetic Algorithms for Military Planning and Scheduling Problems: Applied to Course of Action Planning

**Guitouni, Adel; Belfares, Lamia; DRDC Valcartier TR 2003-372; R & D pour la défense Canada – Valcartier; April 2008.**

Le développement de suites d'actions (COAs) est une entreprise très importante. Les directives et l'intention du Commandant devraient être comprises et traduites concrètement et rapidement en plans complets et flexibles. Les COAs "devraient répondre aux questions fondamentales de quand, qui, quoi, où, pourquoi et comment". Chaque COA devrait être appropriée, réalisable, acceptable, exclusive et complète. Une bonne COA positionne la force pour l'avenir des opérations et fournit la flexibilité pour rencontrer des événements imprévus pendant son exécution. « Qui » dans une COA ne spécifie pas des unités individuelles, mais utilise plutôt des ressources et des capacités génériques. Pendant le développement, le personnel analyse la puissance relative de combat de forces amicales et ennemies et produit des COAs complètes.

Le défi pour les officiers de planification est de produire des plans qui synchronisent des activités et des capacités dans l'espace et dans le temps tout en considérant les interdépendances, les relations de préséance, les contraintes de ressources et la satisfaction de critères multiples, incommensurables et souvent contradictoires. Nous proposons de modéliser une COA comme problème de planification de projet muti-modal avec des contraintes de ressources (MRCPS). D'un point de vue méthodologique, ces deux problèmes ne sont pas tellement différents. Le modèle consiste dans la représentation d'activités génériques (des tâches), avec des relations d'interdépendance entre les activités et nécessitant la combinaison de ressources spécifique de capacités pour leur réalisation. Une action élémentaire est une activité à laquelle une combinaison de ressources ou de capacités a été affectée dans un temps et un espace précis. Nous supposons qu'une mission complexe est décomposable en activités simples granulaires avec des dépendances appropriées. Ce processus implique l'identification des tâches (quoi) aussi bien que leurs rapports de dépendances (quand et où), l'association de capacités disponibles (comment) avec leur localisation et finalement les objectifs de la mission (pourquoi). Une COA est représentée comme un graphe spatio-temporel orienté.

La résolution de la planification de COA est un problème NP-Dur. Cela signifie qu'il est impossible de résoudre un tel problème avec des méthodes exactes. Pour obtenir des alternatives faisables prometteuses en ce qui concerne des objectifs multiples, nous avons pensé à explorer le potentiel des algorithmes évolutionnaires (AEs). Les AEs est une classe de méta-heuristiques qui ont fait leur preuve dans la résolution des problèmes NP-Durs combinatoires.

Dans ce travail, des variantes de COA sont produites utilisant un algorithme évolutionnaire spécifique : les algorithmes génétiques (AGs). Un AG est capable de traiter simultanément des solutions multiples pour résoudre des problèmes d'optimization. Cela permet de trouver des solutions potentiellement non dominées (Pareto optimales) dans la même itération sans aucune sensibilité sur la forme ou la continuité du front Pareto (Coello Coello, 1999).

Les solutions potentiellement Pareto optimales sont obtenues, dans le présent travail, en utilisant des procédures originales basées sur le concept de dominance combiné avec une nouvelle approche de filtrage de multicritère de premier ordre.

Ce travail fait partie du projet 13dm (Systèmes de planification de mission).

This page intentionally left blank.

# Table of contents

# List of figures

# List of tables

# Acknowledgements

The authors thank Dr N. Lo and Mr Z. Malki for their contribution to implement the algorithms. The authors acknowledge Mr. Jean Berger for his technical advices and references as well as the anonymous referee.

# 1 Introduction

The Course of Action (COA) Development step of the Military Operational Planning Process (OPP) involves the entire staff. The Commander's guidance and intent helps the staff to focus on the development of comprehensive and flexible plans within the time available. These COAs "should answer the fundamental questions of when, who, what, where, why and how" [CFC Toronto, 2000, US Army, 1997]. Each COA should be suitable, feasible, acceptable, exclusive and complete. A good COA positions the force for the future operations and provides flexibility to meet unforeseen events during its execution. The "who" in a COA does not specify individual units, but rather uses generic assets and capabilities. During the development step, staffs analyse the relative combat power of friendly and enemy forces, and generate comprehensive COAs.

Staff should identify the required tasks for mission success. These tasks could be decomposed into sub-tasks. Tasks and sub-tasks could be represented by means of a hierarchical structure called the work break-down structure. Leafs of this hierarchical structure are called elementary tasks in this report. Synchronization analyses lead to identifying temporal and spatial relationships between elementary tasks (e.g., End-Start, Start-Start, End-End, Time Lapse, Same Spatial Zone…). Staff should then consider all available resources and capabilities and assign them to the tasks based on capabilities-to-tasks rules. Synchronizing COA requires scheduling starting and ending times of all tasks according to resource availabilities, deployment constraints and task relationships. Any resource or capability has an availability calendar, in-use costing, required preparations, required staff, etc…

In summary, the challenge for the planning officers is to generate complex, spatially and temporally interdependent activities with precedence relationships, subject to resource constraints, and satisfying multiple incommensurable and often conflicting criteria. Guitouni et al. (2000, 2002) proposed to model COA planning as a multiple mode resource-constrained project-scheduling problem (MRCPS). From a methodological point of view, planning and scheduling are quite similar. The model decomposes generic activities (tasks with specific combination of capabilities) into elementary (or simple) interrelated actions to accomplish the mission objectives. We assume that a complex mission is decomposable into granular simple activities with appropriate dependencies. This process implies the identification of the tasks (what) as well as their dependency relationships (when and where), the pool of available capabilities (how) with their location, and finally the objectives of the mission (why). A COA is then represented as an oriented time-space graph as represented by Figure 1. Depending on the combination of resources allocated and action's position in the schedule, different COA networks could be obtained. They constitute variants (or alternatives) of how the mission may be achieved based on different evaluations of objectives.

Solving COA planning problems is NP-Hard. This means that it is impossible to solve such problem with exact methods. To obtain promising feasible alternatives with respect to multiple objectives, we thought to explore the potential of evolutionary algorithms (EAs). EAs are meta-heuristics that have been already proven to be reliable for solving combinatorial NP-hard problems.

Several approaches have been proposed to deal with the multi-objective aspect of resource-constrained project-scheduling (RCPS) problems. The most common one aggregates all

objectives into one single scalar value by using weighted sum, or alike functions, according to the decision maker's preferences set. The search is then performed several times to find a compromise solution that reflects these preferences. Another approach is to generate the set of compromise or non-dominated solutions. This has increased the interest for the Pareto optimization and multi-objective EAs. However, applications to scheduling problems are still scarce (Landa Silva and Burke, 2002). The concepts of Pareto optimization and non-dominated solutions are described in section 2.1.

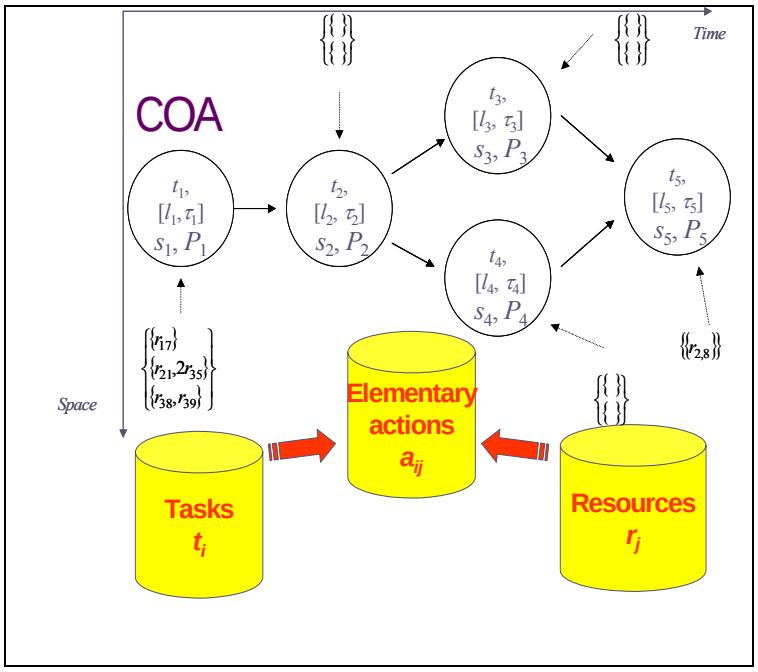


*Figure 1. COA graph representation.*

In this work, different COA variants are generated using a specific evolutionary algorithm: the genetic algorithms (GAs). GAs are able to deal simultaneously with multiple solutions for solving multi-objective optimization problems. This allows finding a set of potential Pareto optimal solutions in the same iteration without any sensitivity on the shape or the continuity of the Pareto front (Coello Coello, 1999). The set of potential Pareto solutions is obtained here by investigating different procedures based on the dominance concept combined with a new approach referred to as 1st order multi-criteria filtering method.

This report is organised as follows. In chapter 2, we present Pareto-optimality. In chapter 3 a review on the multi-objective EA (MOEA) published in the open literature is presented. Chapter 4 is devoted to the multi-objective genetic Algorithm MFGA features being developed for COA planning. Chapter 5 resumes metrics proposed in the literature to evaluate the performance of MOEAs. In chapter 6, computational results of the proposed approach and a comparison with EAs from literature are analysed leading to the conclusions in chapter 7.

This work has been performed under the 13dm project (2002-2003).

# 2 Pareto-based multi-objective optimization

Research on multi-objective optimization goes back to the XIX century with Vilfredo Pareto (1896) who introduced the concept of *Pareto optimum*. Pareto optimality is an optimality criterion for optimization problems with multi-criteria objectives. Multi-criteria optimization is also known as multi-objective optimization, vector optimization, or multi-criteria decision making. The goals are usually conflicting so that an optimal solution in the conventional sense does not exist. Instead one aims at Pareto optimality. A set of object parameters is said to be Pareto optimal, if there is no other state dominating the state with respect to a set of objective functions. A state dominates a state, if is better than in at least one objective function and not worse with respect to all other objective functions.

The multi-objective optimization problem (MOP) can be defined, as proposed by Osyczka (1985):

> "*A vector of decision variables, which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution, which would give the values of all the objective functions acceptable to the decision maker.*"

Formally stated, MOP is formulated as follows: Find the vector $\underline{x}^* = [x_1^*, x_2^*\ldots, x_n^*]$ which satisfies the $m$ inequalities constraints: $g_i(\underline{x}) \geq 0$, $i = 1, 2, \ldots, m,$ the $p$ equality constraints $h_j(\underline{x}) = 0$, $j = 1, 2, \ldots, p,$ and optimizes the vector objective functions $\underline{f}(\underline{x}) = \left[ f_1(\underline{x}), f_2(\underline{x}), ..., f_Z(\underline{x}) \right]^Z$.

The constraints inequalities and equalities define the **feasible region** $\Omega$ and any point $\underline{x}$ in $\Omega$ defines a **feasible solution**. The $Z$ components of the vector function $\underline{f}(\underline{x})$ represent the non-commensurable criteria of the problem which are to be all maximized or all minimized or some to be maximized and others to be minimized. In the later case, a conversion of the functions to be maximized into a minimization form, or inversely, is done before carrying out the optimization. Generally, the solution of a MOP is not unique but is represented by a set of solutions which are referred as good compromises (or trade-offs). This concept of Pareto optimality was originally proposed by Francis Ysidro Edgeworth in 1881 and generalized, later, by Vilfredo Pareto (1896). In a maximization optimization for example, a vector solution x* is called *Pareto optimal* if there exists no feasible vector x which would increase some criterion without causing a simultaneous decrease in at least one other criterion. It is commonly accepted as the **Pareto optimum**. *Pareto* optimum set is generally called an **efficient** solution set or a **non-dominated** solution set.

The set of these efficient solutions in the decision space, $\Omega$, is denoted as the **Pareto set** (noted as $P_{true}$ in this work (Horn, 1997)) and in the objective space, $\Lambda$, as the **Pareto front** ($PF_{true}$).

Dominance relationships between two solutions u and x are defined as follows:

- $x$ absolutely dominates $u$ denoted ($x$ $\mathbf{D}^a$ $u$) $\Leftrightarrow f_i(x) > f_i(u)\ \forall\ i = 1, 2,\ldots, Z,$

- $x$ strictly dominates $u$ denoted ($x \mathbf{D}^s u$) $\Leftrightarrow f_i(x) \geq f_i(u)\ \forall\ i = 1, 2,\ldots, Z$, and $\exists j$ where $f_j(x) > f_j(u)$,
- $x$ weakly dominates $u$ denoted ($x \mathbf{D}^w u$) $\Leftrightarrow f_i(x) \geq f_i(u), \forall\ i = 1, 2,\ldots, Z$.

We have $\mathbf{D}^a \subseteq \mathbf{D}^s \subseteq \mathbf{D}^w$. The set of efficient solutions is also denoted *Pareto set* ($P_{true}$) in the decision space and *Pareto front* ($PF_{true}$) in the objective space.

Generally, in combinatorial optimization, weak dominance is applied because it permits more solutions to be reached from the existing ones with regard to the connectedness of the search space. In NP-hard problems, it is impossible to characterise *a priori* the set of efficient solutions. Therefore, we will use the potential or **approximation efficient set** ($P_{known}$) to represent the **near Pareto optimal solutions**.

The process of a MOP has two stages:

1. Find the Pareto set or well distributed solutions belonging to this set that uniformly cover the Pareto front, and

2. Decide what kind of "trade offs" are appropriate in the decision making (DM) perspective (the so-called multi-criteria decision making process).

The most popular method proposed in the literature to classify the techniques available for these two stages of a multi-objective optimization problem is that of Cohon and Marks (1975):

- Generating techniques (*a posteriori* articulation of preferences): search before making decisions (search $\Rightarrow$ decide)
- Techniques which rely on prior articulation of preferences (non interactive methods): make decisions before searching (decide $\Rightarrow$ search)
- Techniques, which rely on progressive articulation of preferences (interaction with DM): integrate search and decision-making (search $\Leftrightarrow$ decide).

Coello Coello et al. (2002) have summarized the most representative operational research techniques lying in each category. It has applications in a variety of fields, including economics, evolutionary biology, political science, and military strategy. Pareto optimality is a domain-independent property. It is frequently difficult to find a global objective function even for problems that otherwise can be easily mapped into integer programming (IP), a general technique for satisfying multiple objectives with constraints. This is because it is difficult to assign a metric to different objectives. For example, how should the cost of the artefact be weighted with respect to its risk or various features? The difficulty of generating an "objective" function is increased by the fact that even small numeric changes in weights can generate very different solutions.

No part of a Pareto optimal solution can be improved without making some other part worse. Pareto optimality is a predicate. While one may be able to assign a quantitative metric, the answer as to whether the global solution is Pareto optimal is "yes" or "no". A corollary is that Pareto optimality does not address local extrema with respect to any utility. Neither does Pareto optimality provide a method for choosing among preferences or alternatives. Nevertheless, tracking Pareto optimality is an important function. Detection of a lack of Pareto optimality is an

alert to an opportunity to improve the design that otherwise might be missed, especially when no one staff officer understands all of the design dependencies. Once such a lack has been detected, then special purpose algorithms can provide various evaluation functions that are likely to be domain-specific. Tracking Pareto optimality does not preclude such methods and it does not require an objective function that must compare “apples and oranges” in complex domains: it is a domain-independent function.

# 3 Evolutionary algorithms for multi-objective optimization problems: A review

This section presents a quantitative and qualitative analysis of currently known published research on multi-objective evolutionary algorithms (MOEAs). This review is based essentially on the book of Coello Coello et al. (2002) which have done a remarkable job in collecting, organizing and interpreting the burgeoning literature of MOEAs. Let's consider the following notations:

- $P_{current}(t)$ : non dominated solutions in the current population (at generation t) in the decision variable space
- $PF_{current}(t)$ : non dominated solutions in the current population (at generation t) in the objective space
- $P_{known}(t)$ : non dominated solutions found through the generations or approximation efficient set in the decision variable space
- $PF_{known}(t)$ : non dominated solutions found through the generations or approximation efficient set in the objective space
- $P_{true}$ : Pareto optimal set of solutions in the decision variable space
- $PF_{true}$ : Pareto optimal set of solutions in the objective space
- t : generation number
- $t_{dom}$ : cardinality of the comparison set in NPGA method
- $\sigma_{share}$ : size of the niche

## 3.1 Evolutionary Algorithms (EAs) basics

This section presents basic EA structural terms and concepts referenced in the book of Coello Coello et al. (2002). The reader is referred to Goldberg (1989), Michalewicz (1996) or Mitchell (1996) for a general description and to Bäck (1996) for a more technical presentation.

There is no universally accepted definition of EA, but in a strict sense an evolutionary algorithm handles a population of solutions, evolves this population by means of cooperation (recombination) and self-adaptation (mutation) and uses a coded representation of solutions (Hertz and Klober, 2000). Although Genetic programming and learning classifier may be considered as evolutionary techniques, several researchers consider them conceptually different from evolutionary computation (Kinnear et al., 1997). Genetic algorithm is a particular technique of EAs.

The three features which characterize EAs are:

– Encoded solutions: A solution to some problem is encoded and called a **structure** or an **individual**. An individual is typically represented as a string (or string of strings) corresponding to a biological **genotype**. When decoded it is expressed into a **phenotype** (solution). A genotype is composed of one or more **chromosomes** (strings or matrix), where

each chromosome is composed of **genes** (an element of a string). These genes are characterized by a position in the chromosome, **the locus**, and take on certain values (real, binary, qualities…): **the alleles**.

– Fitness: An EA requires both an objective and fitness functions, which are fundamentally different, to be assigned to an individual for its evaluation. The objective function defines the EA's optimality condition (and is a feature of the problem domain) while the fitness (in the algorithm domain) measures the desirability of an individual.

– Operators: Just as in nature, evolutionary operators (EVOPs) operate on a population attempting to generate solutions with higher and higher fitness. The three major EVOPs are **mutation**, **recombination** and **selection**. The EVOPs are applied on solutions with a probability proportional to their fitness. The mutation consists of changing the allele or the locus of a gene in a chromosome. For example, change, in a binary coded string, "1" to "0", or select two positions in a (integer or alphabet coded) string and swap the genes on these positions. The recombination consists of exchanging information (pieces of chromosome) between individuals to generate new ones or generating a new solution by combining the genes (or pieces) of several individuals. Individuals in the population are selected (reproduced) to become members of the next generation. The individuals generated from these two operators are called **children** or **offspring**. The selection EVOP gives individuals with higher fitness a higher probability of contributing with one or more children in the next generation.

Various versions of EVOPs are proposed in the literature. They are often adapted to the problem domain constraints that affect chromosome structure and alleles (Bäck, 1996). EA selection technique is the control mechanism determining the type of search performed. Bäck's analysis shows a general ordering of selection techniques (listed in order of increasing selective procedure): proportional, linear ranking, tournament, and $(\mu,\lambda)$ selection (Bäck, 1996, pg.186). Tournament selection operates by randomly choosing k individuals from the generational population and selecting the "best" to survive in the next generation. Binary tournament (k=2) are probably the most common. Ranking assigns selection probability solely on individual's rank, ignoring the absolute fitness values. $(\mu+\lambda)$ selection strategy, where $\mu$ represents the number of parents and $\lambda$ the number of children, select the $\mu$ best individuals drawing from both the parents and children and in $(\mu,\lambda)$ strategy, the $\mu$ best individuals ($\mu<\lambda$) are selected from the child population only.

Three major instantiations exist in evolutionary computation (EC) and the differences between them are highlighted on Table 1 (Coello Coello et al. 2002). Interested readers are directed to the Handbook of evolutionary computation (Bäck et al., 1997) for an in-depth analysis of general EVOPs and EA components. This book is the most comprehensive collection of articles discussing EC, its instantiation and applications.

As pointed out in the book of Coello Coello et al. (2002), although a large field for creativity exists when defining EA instantiations (such as genetic representation and specific EVOPs) careful consideration must be given to the mapping from problem to algorithm domains. Inappropriate representations and/or EVOps may lead to a bad EA performance (e.g. Hamming cliffs (Bäck, 1996, pg. 229)). There is no unique combination guaranteeing successful EA utilization (Fogel and Ghozeil, 1997; Wolpert and Macready, 1997) but choosing wisely may well result in more effective and efficient implementations.

*Table 1. Key EA implementation differences*

| EA Type | Representation | EVOPs |
|---|---|---|
| Evolutionary Programming (EP) | Real-values | Mutation and (μ+λ) selection alone |
| Evolutionary strategy (ES) | Real-values and strategy parameters | Mutation, recombination and (μ+λ) or (μ,λ) selection |
| Genetic algorithms (GA) | Historically binary; Real values now common | Mutation, recombination and selection |

The formal definition of an EA, as given by Merkle and Lamont (1997) (see Bäck, 1996, pg.66), is:

- Let I be a non-empty set (the individual space), $\{\mu^{(i)}\}_{i \in N}$ a sequence in $Z^+$ (the parent population sizes), $\{\mu'^{(i)}\}_{i \in N}$ a sequence in $Z^+$ (the offspring population sizes), $\Phi : I \rightarrow R$ a fitness function, $\iota: \bigcup_{i=1}^{\infty} (I^{\mu})^{(i)} \rightarrow \{true, false\}$ }, r a sequence $\{r^{(i)}\}$ of recombination operators $r^{(i)} : X_r^{(i)} \rightarrow \Gamma\left(\Omega_r^{(i)}, \Gamma(I^{\mu(i)}, I^{\mu'(i)})\right)$, m a sequence $\{m^{(i)}\}$ of recombination operators $m^{(i)} : X_m^{(i)} \rightarrow \Gamma\left(\Omega_m^{(i)}, \Gamma(I^{\mu'(i)}, I^{\mu'(i)})\right)$, s a sequence $\{s^{(i)}\}$ of selection operators $s^{(i)} : X_s^{(i)} \times \Gamma(I, R) \rightarrow \Gamma(\Omega_s^{(i)}, \Gamma\left((I^{\mu'(i)+\chi\mu(i)}), I^{\mu(i+1)}\right)$, $\Theta_r^{(i)} \in X_r^{(i)}$ (the recombination parameters), $\Theta_m^{(i)} \in X_m^{(i)}$ (the mutation parameters) and $\theta_S^{(i)} \in X_s^{(i)}$ (the selection parameters).
- Then the algorithm shown in Figure 2 is called an Evolutionary Algorithm.

$t := 0$

$initialize P(0) := \{a_1(0),...,a_\mu(0)\} \in I^{\mu(0)};$

**While**$(\iota(\{P(0),...,P(t)\} \neq true)$ **do**

Re*combine*: $P'(t) := r^{(t)}_{\Theta_r{}^{(t)}}(P(t));$

*Mutate*: $P''(t) := m^{(t)}_{\Theta_m{}^{(t)}}(P'(t));$

*Select*:

**if** $\chi$

**then** $P(t+1) := s^{(t)}_{(\theta_s^{(t)},\phi)}(P''(t));$

**else** $P(t+1) := s^{(t)}_{(\theta_s^{(t)},\phi)}(P''(t) \cup P(t));$

**endif**

$t := t+1$

**endwhile**

*Figure 2. Evolutionary algorithm outline*

## 3.2 MOEA Classification

Many MOEA approaches owe their success to previously mathematical MOP techniques. These MOP techniques are themselves related to the operational research (OR) field methods (Cohon and Marks, 1975; Hwang and Masud, 1979; Steuer, 1986) as their multiple objective decision making (MODM) problems have similar characteristics. These are a set of quantifiable objectives, a set of well-defined constraints, and a process of obtaining trade-off information between the stated objectives (and possibly also between stated and non-stated non-quantifiable objectives) (Hwang and Masud, 1979).

Various MODM techniques are commonly classified from a DM's point of view (i.e. how the DM performs search and decision making). Cohon and Marks (1975) further distinguish between a single DM/group or multiple DMs with conflicting decisions. Generally, in MOEA approaches, a single DM or a group united in its decisions are considered. Moreover, as mentioned in section 2, the final MOP solution(s) results from both optimization and decision processes. The final solution(s) results from a DM's preferences being made known before, during, or after the optimization. Coello Coello et al. (2002) adopt the same classification (Figure 3) for MOEA-based MOP solutions techniques in their review.

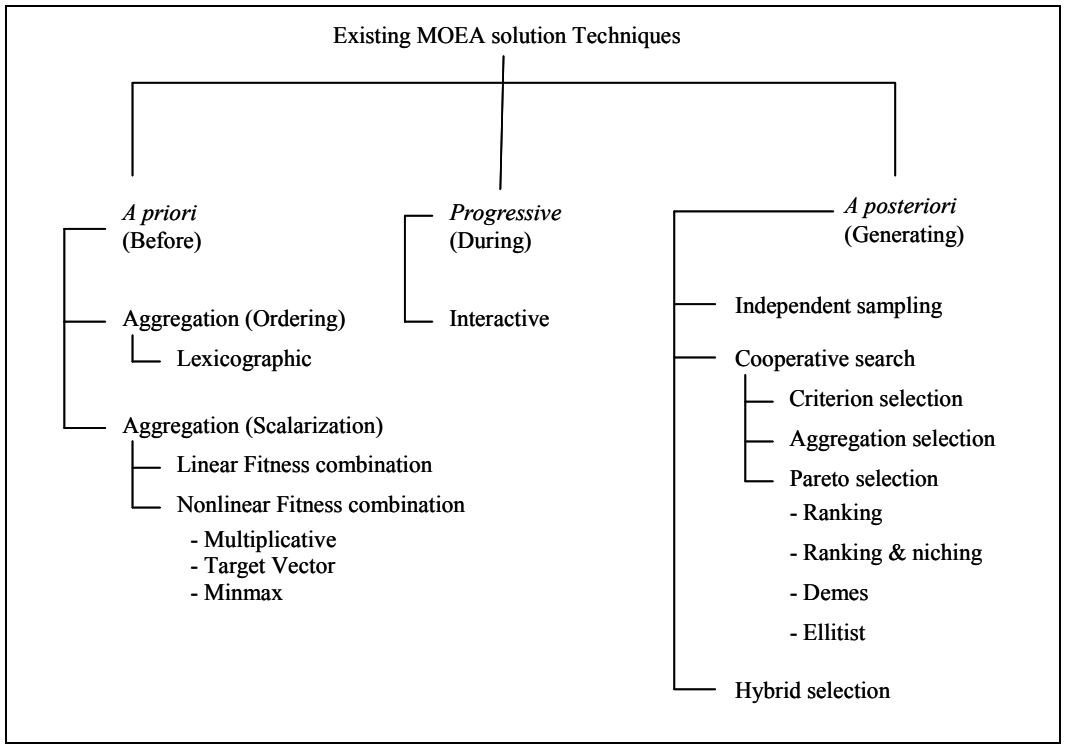


*Figure 3. MOEA solution technique classification (Coello Coello et al. 2002)*

## 3.3 MOEA techniques

This section addresses the many MOEA techniques proposed in the literature and reviewed by Coello Coello et al. (2002). Each citation is presented by underlying key elements of its approach and classified using the structure defined in Figure 2. For a detailed survey of EA techniques and their applications, the reader is referred to the book of Coello Coello et al. (2002).

### 3.3.1 *A priori* techniques

By definition, these techniques require the definition of objective importance before search occurs and this is not a trivial task in real-world scientist and engineering problems. Depending on the objective prioritization chosen, The DM's "cost" (no matter how defined) could be greater than necessary, missing in this way more "acceptable" solutions. No matter, the optimization algorithm used, this is an unavoidable consequence of *a priori* MOEA techniques that can be categorized in following:

- Lexicographic ordering
- Linear aggregating functions
- Nonlinear aggregating functions

**Lexicographic Ordering**: In this method, the objectives are ranked in order of importance by the DM. A first solution $\underline{x}^*$ is then obtained by minimizing the objective function starting with the most important one and proceeding according to the assigned order of importance of the objectives. Let $f_1(\underline{x})$ and $f_z(\underline{x})$ denote the most and least important objective functions, respectively. Then the first problem is formulated as

$$\text{Minimize } f_1(\underline{x}) \qquad \text{s.t.} \qquad g_i(x) \leq 0 \quad i = 1,2,...,m \tag{1}$$

and its solution is $\underline{x}_1^*$ and $f_1^* = f_1(\underline{x}^*)$ is obtained. Then the second problem is formulated as:

$$\text{Minimize } f_2(\underline{x}) \qquad \text{s.t. } g_i(x) \leq 0 \quad i = 1, 2,..., m. \tag{2}$$

$f_1(\underline{x}) = f_1^*$ and the solution of this problem is obtained as $\underline{x}_2^*$ and $f_2^* = f_2(\underline{x}^*)$.

This procedure is repeated until all $Z$ objectives have been considered. The solution obtained at the end ($\underline{x}_Z^*$) is considered as the desired solution of the problem. It is also possible to select randomly an objective to be optimized during iteration (Fourmen, 1985).

This method is equivalent to a weighted combination of objectives where each weight is defined in terms of the probability that each objective has of being selected. Its main weakness is that this approach will tend to favour more certain objectives to the detriment of others. The search procedure makes the population to converge towards a particular part of the Pareto front rather than to delineate it completely (Coello Coello, 1996). The main advantage of this method is its simplicity and computational efficiency. This makes it highly competitive with other non-Pareto methods such as a weighted sum of objectives or VEGA (see Criterion selection techniques). As pointed out by Coello Coello et al. (2002), the lexicographic technique appears most suitable only when the importance of each objective (in comparison to the others) is clearly known. The search space is explored in the sense that priority is given to solutions performing well in one objective over other(s). Thus, one objective is optimized at all costs. In this case it seems more appropriate to use instead a single objective EA, which does not incur the additional overhead of a MOEA!

– **Linear aggregating functions**: The linear aggregation approach converts the MOP into a single objective problem by computing the fitness of each individual using:

$$\text{fitness} = \sum_{i=1}^{Z} \omega_i f_i(\underline{x}) \tag{3}$$

where $\omega_i \geq 0$ are the weights coefficients representing the relative importance of the Z objectives of the problem. It is usually assumed that

$$\sum_{i=1}^{Z} \omega_i = 1 \tag{4}$$

The weights could be also varied in order to delineate the Pareto front. This technique is popular despite its identified shortfalls (Das and Dennis, 1997), probably due to its simplicity. The reader can consult the many real-world applications presented in the book of Coello Coello et al. (2002).

A basic weighted sum MOEA is easy to understand and implement; the fitness combination technique is also computationally efficient. However this technique presents a serious difficulty when the objectives are non-commensurable or simply are not in the same scale. The normalization of the functions is then unavoidable. This implies that we should know, to a certain extent, the behaviour of each objective, which is (in most real-world applications) a very expensive process. Another important disadvantage is that this approach does not generate proper Pareto optimal solutions in the presence of non-convex search spaces (Das and Dennis, 1997).

– **Nonlinear aggregating functions**: Nonlinear aggregating functions are not very popular due to the difficulties encountered in determining appropriate information involved in the fitness construction. This is the example of the utility function method (Tabucanon, 1989). This technique converts the MOP into a single objective problem where the utility function, F, of the multiple objectives represents the DM's preferences. The construction of F can be extremely difficult due to the determination of appropriate probabilities of acceptance and to the conditions that the objective functions must meet (Keeney and Raiffa, 1976).

A more popular and useful approach, when the desirable goals are known, is the target vector technique. The fitness of the EA, to be minimized, measures the absolute deviation of the current solution from the vector of desirable goals. The most used techniques are hybrids with goal programming (Wienke et al., 1992; Sandgren, 1994; Deb, 1999a), goal attainment (Wilson and McLeod, 1993; Zebulum et al., 1998), game theory (Rao, 1993; Dhingra and Lee, 1944; Sefrioui and Periaux, 2000; Wang and Periaux, 2001) and the min max algorithm (Hajela and Lin, 1992; Coello Coello and Christiansen, 1998). The formulation of the fitness in the target vector approach is as follows (Coello Coello et al. 2002):

$$\text{fitness} = \left\| [f(\underline{x}) - g] W^{-1} \right\|_{\alpha} \tag{5}$$

Where $g = [g_1, g_2, ..., g_Z]$ is the vector of desired goals. W is a weighting matrix accounting differing variance between the goals and $\alpha$ is most often the Euclidian distance ($\alpha = 2$). If $\alpha = \infty$, the above equation is equivalent to the min max function. This technique is mathematically represented by:

$$\text{fitness} = \max_{j=1,...,Z} \frac{f_j(\underline{x}) - g_j}{\omega_j} \tag{6}$$

Minmax is a scalar aggregative method minimizing the maximum (weighted) difference between the objectives and DM-specified goals.

As mentioned above, the non-linear aggregating methods are easy to implement but present some difficulties in defining the fitness functions. These techniques are high level search space knowledge demanding. Goal attainment technique has several weaknesses as indicated by Wilson

and Macleod (1993) from which the main one is the misleading pressure selection generated under certain circumstances. For example, if two solutions have in one objective the same function value but different values in the other, they still have the same goal attainment value for their two objectives, which means that for the EA neither of them is better than the other. An additional problem with target vector approaches is that they yield non dominated solutions only if the goals are chosen in the feasible domain. If the feasible region is difficult to approach, the method becomes inefficient. These weaknesses are added to the disadvantages associated with the aggregative methods (Coello Coello, 1999).

However, for certain combinatorial optimization problems, nonlinear aggregating functions (e.g. based on Tchebycheff weights) can provide very good approximation of the Pareto optimal set (see for example Jaszkiewicz, (2000, 2001); Jaszkiewicz et al. (2001)).

### 3.3.2 Progressive techniques

Coello Coello et al. (2002) notice, with surprise, in their review on MOEAs, the lack of cited interactive search efforts in this domain. It seems intuitive that a closer collaboration, between the DM and "searchers", can only lead to a better quality of the results. Moreover, in real-world applications, this interactive process implies significant economic improvements. Although *a priori* or *a posteriori* techniques could be used in the interactive process, the authors suggest that the latter are more appropriate because they generate a set of solutions rather than one. They advance that some of these *a posteriori* methods are able to explicitly incorporate DM preferences within the search (Fonseca and Flemming, 1993; Esbensen and Kuh, 1996; Hu et al. 1996; Greenwood et al., 1997; Cvetković and Parmee, 1999; Rekiek, 2000).

**Strengths and weaknesses:** Similarly to the *a priori* techniques, the interactive approach necessitates knowledge of the problem domain in order to bias the search in the direction of the defined goals and according to the scheme of preferences. Moreover, difficulties may arise in cases where there are contradictions (when dealing with non-unified group preferences). However, when the search is constrained within a certain region of interest, an interactive process may be the best choice (Coello Coello et al., 2002).

### 3.3.3 *A posteriori* techniques

These techniques perform an MOP search process attempting to identify the set of "all" possible optimal solutions denoted by $P_{true}$. Generally, an MOEA execution, for a particular instance of problem, results in an *approximation Pareto set* of solutions $P_{known}$ and their evaluations $PF_{known}$. Then, depending upon the size of this set, an interaction process with the DM is performed to decide on which solutions will be retained. As summarized on Figure 2, *a posteriori* techniques are based on three approaches:

- Independent sampling
- Cooperative search
- Hybrid Selection

**Independent sampling**: This technique uses multiple single-criterion searches. EAs are executed several times with different sets of weights to discover the greater number of non dominated

solutions and ultimately $P_{true}$. A few independent sampling approaches are reported but their overall effectiveness is in doubt (Coello Coello et al., 2002).

This is same as the linear aggregating techniques. However, as pointed out by Coello Coello et al. (2002), this method could be useful to approximate convex Pareto front in multi-objective combinatorial problems. In this case they may produce competitive results with respect to other MOEAS based on Pareto ranking at a lower computational cost (Srigiriraju, 2000).

**Criterion selection techniques**: These techniques are based on the idea that the selection is performed independently for each criterion and the vector evaluated genetic algorithm (VEGA) proposed by Schaffer (1985) is an example. It is probably the first genetic algorithm in which the concept of dominance was implemented to evaluate and select individuals. In each generation, a group of individuals is selected according to one of the Z objective functions in the problem until Z groups of equal size, N/Z, are formed if N is the size of the population. Then the Z groups are shuffled together and the genetic operators are applied to form new population. This is done in order to achieve mating among individuals belonging to different sub-groups, combining, by this way, part of strings that confer higher fitness. A problem with this scheme as pointed out by Schaffer is that the selection procedure is biased against members that are not excellent along any criterion which have "middling" performance in all dimensions. These solutions could be very useful for compromise solutions but are unable to survive with the VEGA selection scheme. This is a consequence of what is called in genetics, "the speciation", i.e. evolution of "species" that excel on one performance without looking at others. To overcome this difficulty, Schaffer introduced several transformations such as a redistribution scheme and a crossbreeding plan (mating restrictions) but none appear to be effective.

Norris and Crossley (1998) and Crossley et al. (1998) have implemented elitism to ensure the $PF_{known}(t)$ endpoints (extrema) survive between generations. To preserve diversity, they also introduce Z-branch tournament, (where Z is the number of objectives) (Jones et al., 1998). This allows each solution to compete once in each of Z tournaments, where each set of tournaments selects 1/Z of the next population.

VEGA is very simple and easy to implement and can generate several solutions in one run. However, shuffling and merging all subpopulations corresponds to averaging the fitness components associated with each of objectives (Goldberg and Richardson, 1987). Since the technique uses proportional fitness assignment (Goldberg, 1989), these fitness components are in turn proportional to the objectives themselves (Fonseca and Flemming, 1995). Richardson et al. (1989) showed that the resulting expected fitness corresponds to a linear aggregation of the objectives where the weights depend on the distribution of the population at each generation. This means that VEGA has the same problems as the aggregation techniques. Nevertheless, VEGA has been found useful in certain domains such as constraint handling, where its biased behaviour can be of great help (Surry et al., 1995; Coello Coello et al., 2000).

## 3.4 Aggregation selection techniques

Aggregative selection MOEAs incorporate a variety of methods such as weighted sums (Ishibuchi and Murata, 1998), constraints and objective combinations (Loughlin and Ranjithan, 1997), and hybrid search approaches (Deb, 1999b). In these techniques, the weights are varied between the

generations and/or each function evaluation. Sometimes they are assigned randomly; sometimes they are optimized through a function or encoded in the chromosomes as genes where the EVOPs act upon them also.

The main advantage of these techniques, similarly to the criterion selection techniques, is that they can generate a set of solutions in a single run of MOEA. That is, an approximation Pareto set could be obtained. However, these techniques are not without disadvantages that parallel those of weighted sums (Das and Dennis, 1997) such as missing certain members of $PF_{true}$. Furthermore, both the constraints/objectives combination and hybrid search methods have significant overhead (e.g. solving nonlinear system of equations to determine an appropriate hyperplane (Zhou and Gen, 1997)).

**Pareto sampling**: The Pareto sampling techniques offer the capability to find all members of $P_{true}$ and $PF_{true}$. This is done by using Pareto-based fitness assignment to identify non dominated solutions from the current population. Pareto concepts are used in the selection procedure such that Pareto solutions are given preference over dominated solutions, but are treated equivalently among themselves. Several methods of Pareto sampling exist and succinct descriptions follow.

**Ranking:** The Pareto-based fitness assignment was first proposed by Goldberg (1989) to solve the problems of Schaffer's approach (VEGA (Schaffer, 1985)). He suggested the use of non dominated sorting to accord the same *reproductive potential* to the "middling" solutions (same probability of getting selected to propagate genetic material to progeny) as those near the extremes. Specifically, he said that the population should be ranked on the basis of non domination. For this, all non dominated individuals in the current population are to be identified and flagged. These individuals are to be placed at the top of the list and assigned rank of 1. These points are then removed from contention and the next set of non dominated individuals are identified and assigned rank 2. This process is to be continued until the entire population is assigned a rank. Others ranking schemes were later proposed such as Fonseca and Flemming's technique, and Zitzler and Thiele's technique.

Fitness is associated to each rank and the probability of selection of an individual is then calculated based on this fitness. So, the population is subdivided into sets having the same rank. This type of blocked fitness assignment produces large selection pressure that might induce premature convergence. To avoid this and maintain the diversity of the population, the concept of niche, species and sharing functions were introduced.

**Niching and fitness sharing**: Deb and Goldberg (1989) find a way to avoid the convergence to a single non-dominated solution, called "genetic drift", the phenomenon in which the simple GA converges towards a single peak of a multi-modal function being optimized. They investigated two niching methods: crowding and sharing.

**Crowding**, suggested by De Jong (1975), creates separate niches by replacing existing members in a niche according to their similarity with other members in an overlapping population. **Sharing** is the other approach to form niches and it occurs in nature when certain resources are shared within an environmental recess. In its GA analog, the shared resource is a member's fitness. To induce sharing in GA (a concept attributed to Holland (1975)) a member's fitness is reduced by an amount proportional to the nearness of others members in its neighbourhood.

One sharing scheme, tested by Deb and Goldberg (1989), was proposed originally by Goldberg and Richardson (1987) for single objective optimization GA. The population is divided into subpopulation according to the similarity of the individuals. Similarity is controlled by comparing the distance, dist($x_i$,$x_j$), of individual members from each other, to a parameter called $\sigma_{share}$. How much sharing will occur between two members is determined by a power law sharing function Sh(dist($x_i$,$x_j$)):

$$Sh(d(x_i,x_j)) = 1 - \left( \frac{dist(x_i, x_j)}{\sigma_{share}} \right)^{\alpha} \quad \text{if } dist(xi,xj) < \sigma_{share}, \text{ and} = 0 \text{ otherwise} \tag{7}$$

$\alpha$ is the power exponent and $\sigma_{share}$ the size of the niche. $\sigma_{share}$ is of utmost importance as it controls the width of the neighbourhood and the maximum distance between members who will share fitness and induced to form a niche.

The fitness of an individual is then modified as:

$$f_s(x_i) = \frac{fitness(x_i)}{\sum_{j=1}^{N} Sh(dist(x_i, x_j))} \tag{8}$$

Deb and Goldberg (1989) proposed a way of estimating the parameter $\sigma_{share}$ in both decision variable space (phenotype) and gene space (encoded solutions). The phenotype sharing distance is simply the Euclidian distance in n-dimensional space, where n is the vector decision dimension (Eq. 1); the value of dist($x_i$,$x_j$) is calculated as:

$$dist(x_i,x_j) = \sqrt{\sum_{k=1}^{n} \left( x_{k,i} - x_{k,j} \right)^2} \tag{9}$$

and the value of $\sigma_{share}$ :

$$\sigma_{share} = \frac{r}{\sqrt[n]{q}} = \frac{\sqrt{\sum_{k=1}^{n} (x_{k,i} - x_{k,j})^2}}{\sqrt[n]{2q}} \tag{10}$$

where r is the volume of a n-dimensional hyper sphere of radius $\sigma_{share}$ and q is the number of peaks that EA aims to find. The sharing parameter may be calculated in the objective space by replacing the decision variable vector by the objective function vector.

In genotype sharing, dist($x_i$,$x_j$) is defined as the Hamming distance between the strings and $\sigma_{share}$ is the maximum number of different bits allowed between the strings to form separate niches in the population. The experiments performed by Deb and Goldberg (1989) showed that sharing is a better way of keeping diversity than crowding , and indicated that phenotype sharing was better than genotype sharing. Several other proposals exist; see Mahfoud and Goldberg (1995) for a more detailed review of approaches to keep diversity. However, MOEAs use sharing in an attempt to find, not a *multiple* optima, but a uniform (equidistant) distribution of vectors representing $PF_{true}$, i.e., one in which $PF_{known}$'s shape is a good approximation of $PF_{true}$. Different restriction sharing schemes are introduced in MOEAs as it can be seen in the Pareto techniques**.**

Most of these methods require setting explicit values for the key sharing parameter $\sigma_{share}$ which can affect both MOEA efficiency and effectiveness. Fitness sharing's performance is also sensitive to the population size N. Assigning appropriate values to $\sigma_{share}$ requires some *a priori* knowledge about the shape and separation of a given problem's niches (Coello Coello et al, 2002).

**Mating restrictions:** Adding mating restriction to promote speciation was another aspect investigated by Deb and Goldberg (1989). Goldberg (1989) first mentions its use in single-objective optimization GA to minimize "low-performance offspring (lethals)". Restricted mating biases how solutions are paired for recombination, in order to increase algorithm effectiveness and efficiency.

Deb and Goldberg (1989) suggested restricting mating between members belonging to different species. Two members are allowed to reproduce only if they are very similar (i.e. if their phenotype distance is less than a factor called $\sigma_{mate}$). *Island model* GAs also implements restricted mating but in geographic sense where solutions mate only with neighbors residing within some restricted topology (Cantú-Paz, 1997).

In MOEAs, however defined, restricted mating is incorporated to reduce unfit offspring. For example, Baita et al. (1995), and Loughlin and Ranjithan (1997) place solutions on a grid and restrict the area of mating. Lis and Eiben (1996), allow mating only between solutions of different "sexes". The number of sexes equals the number of objective functions. Others formulations are reviewed by Coello Coello et al. (2002).

Coello Coello et al. (2002) reported that no clear quantitative evidence regarding restricted mating's benefits exists. The empirical evidence presented in the literature can be interpreted either for or against this feature of MOEAs. They suggested that this issue will benefit from experiments directly comparing its algorithmic inclusion/exclusion. One must also consider the "No Free Lunch" theorems of Wolpert and Mcready (1997) realizing that mating restriction may not always be effective (or needed) for every problem (class).

## 3.5 Pareto techniques in the literature

### 3.5.1 Multi-objective genetic algorithm (MOGA)

The multi-objective genetic algorithm developed by Fonseca and Flemming (1993) is a variation of Goldberg's technique for finding $P_{true}$ and $PF_{true}$. In MOGA, the rank of an individual

corresponds to the number of solutions in the current population by which it is dominated. The non-dominated solutions are assigned rank 1. For example, an individual $x_i$ at generation t, which is dominated by $p_i(t)$ individuals, has a rank given by:

$$\text{Rank}(x_i, t) = 1+p_i(t) \quad (11)$$

Then fitness is assigned to individuals by the following way:

- Sort population according to rank.
- Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $R \leq N$) in the way proposed by Goldberg (1989), according to some function, usually linear, but not necessarily.
- Average the fitness of individuals with the same rank, so that all of them are sampled at the same rate. This procedure (sharing niching) keeps the global population fitness constant while maintaining appropriate selective pressure.

The sharing is performed on the objective function values (Srinivas and Deb, 1995) to avoid large selection pressure that might produce premature convergence. MOGA has also hybridized with neural networks in attempt to improve its performance (Duarte et al., 2000). The MOGA pseudo code, as reported by Coello Coello et al. (2002), is:

```
Initialize population
Evaluate objective values
Assign rank based on Pareto dominance
Compute niche count
Assign linearly scaled fitness
Assign shared fitness
For i = 1 to Genmax
      Selection via stochastic universal
      sampling
      Single point crossover
      Mutation
      Evaluate objective values
      Assign rank based on Pareto dominance
      Compute niche count
      Assign linearly scaled fitness
      Assign shared fitness
End Loop
```

*Figure 4. MOGA pseudo code*

A more recent version of this algorithm is described and compared against other methods in Purshouse and Flemming (2001).

**Strengths and weaknesses:** The main criticism towards MOGA has been that it performs sharing on objective value space, which implies that two different decision vectors with the same objective function values cannot coexist in the population (Srinivas and Deb, 1995; Deb, 1998). However, nothing in the algorithm prevents performing sharing in the decision variable space. The main advantage of MOGA is that it is efficient and relatively easy to implement (Coello Coello et al. 1996; Van Veldhuizen, 1999). Its main weakness, as with all other Pareto techniques, is that its performance is highly dependent on appropriate selection of the sharing factor $\sigma_{share}$. However, it is important to notice that the authors developed later a good methodology to compute this parameter (Fonseca and Flemming, 1998).

### 3.5.2 Niched Pareto genetic algorithm (NPGA)

In this algorithm, Horn and Nafpliotis (1993) proposed a tournament scheme based on the concept of dominance. Two individuals randomly chosen, competing for selection, are compared against a subset of the population ($t_{dom}$ is typically around 10% of the population). If one of them is dominated (by the individuals randomly chosen from the population) and the other is not, the non-dominated individual wins. If both competitors are dominated or non-dominated (i.e. there is a tie), the result of the tournament is decided through fitness sharing. Horn et al. (1994) suggested a form of fitness sharing with the use of a metric that combines both the objective and the decision variable domains leading to what they called *equivalence class sharing*. This is achieved by simply counting the number of points in the population within a certain distance $\sigma_{share}$ from that individual. The individual with the smaller niche count (i.e lower phenotypic neighbours) is then selected.

Later, Erickson et al. (2001) proposed the NPGA2, which uses Pareto ranking besides the niched Pareto tournament. Niche counts in this improved algorithm are calculated using individuals that are already selected as candidates of the next generation rather than using the current population. This is called continuously updated fitness sharing and was proposed originally by Oei et al. (1991). The pseudo code of NPGA2, as reported in Coello Coello et al. (2002), is given by:

```
Initialize population
Evaluate objective values
For i = 1 to Genmax
     Specialized binary tournament selection
     Using degree of domination as rank
     Only candidate 1 dominated : select candidate 2
     Only candidate 2 dominated : select candidate 1
     Both candidates dominated or both not dominated:
     Perform specialized fitness sharing
     Return candidate with lower niche count
     Single point crossover
     Mutation
     Evaluate objective values
End Loop
```

*Figure 5. NPGA2 pseudo code*

**Strengths and weaknesses:** Since NPGA does not apply Pareto selection to the entire population, but only to a part of it ($t_{dom}$ individuals), at each run, its main strengths are that it is very fast and produces good non-dominated fronts that can be kept for large number of generations (Coello Coello, 1996). However the efficiency of the method depends on the sharing factor $\sigma_{share}$ and the size of the tournament $t_{dom}$. The introduction of the Pareto domination sampling in NPGA2 for the tournament makes it less noisy and improves its efficiency (Erickson et al., 2001).

### 3.5.3 Non-dominated sorting genetic algorithm (NSGA):

Srinivas and Deb (1995)'s algorithm is another variation of Goldberg's approach. It is based on several layers of classifications of the individuals. The first "wave" or blocks of individuals that are non-dominated are assigned rank 1 with a dummy fitness value proportional to the population size (to provide an equal reproductive potential). To maintain the diversity, these individuals share their dummy fitness. Then, a second block of non dominated individuals (after removing the first block from the population) are considered. Once the whole population is classified, a stochastic remainder proportionate selection is applied to ensure that the individuals in the first front get more copies for reproduction than the rest of population. An updated version of this algorithm, called NSGAII (Deb et al., 2000a; 2000b) which is more efficient (computationally) uses elitism and crowding that keeps diversity without specifying any additional parameters. The pseudo code of this technique, as reported by Coello Coello et al. (2002), is:

```
Initialize population
Generate random population size N
     Evaluate objective values
     Assign rang (level) based on Pareto dominance-“sort”
Generate child population
     Binary selection tournament
     Recombination and mutation
For i = 1 to number of generation
     With parent and child population
          Assign rank level based on Pareto-“sort”
          Generate sets of non-dominated fronts
          Loop (inside) by adding solutions to next
          generation
          starting from the first front until N
          individuals found
          determine crowding distance between individuals
          on each front
     Select points (elitist) on the lower front and are
     outside the crowding distance
     Create next generation
          Binary tournament selection
          Recombination and mutation
          Increment generation index
End Loop
```

*Figure 6. NSGA pseudo code*

**Strengths and weaknesses:** NSGA: some researchers have reported that NSGA has a lower overall performance than MOGA and seems to be more sensitive to $\sigma_{share}$ (Coello Coello, 1996; Van Veldhuizen, 1999).

**NSGAII**: is more efficient than its previous version, but as reported by Coello Coello and Toscano Pulido (2001), it has questionable exploratory capacity. Although this algorithm tends to spread quickly and appropriately when a certain non-dominated region is found, it seems to have some difficulties in generating non-dominated solutions in certain (isolated) regions of the search space. Zitzler et al. (2001) noticed also a search bias when the number of objectives increases.

### 3.5.4 Strength Pareto evolutionary algorithms (SPEA)

This algorithm was proposed by Zitzler et al. (1999) as an approach that incorporates several of the desirable features of other MOEAs. These features are: the use of dominance to evaluate and select solution, the use of additional population to store non-dominated solutions (elitism), and the use of a niching or clustering scheme called “average linking method” to keep diversity. At each generation, for each non-dominated solution in the external list, a strength value is computed. This strength, proportional to the number of solutions to which a certain individual dominates, is similar to the ranking method in MOGA. The fitness of each member is computed according to the strengths of all external non-dominated solutions that dominate it. This algorithm

was also revised (Zitzler et al. 2001) called SPEA2. SPEA2 has three main differences with respect to its predecessor (Zitzler et al. 2001): 1) it incorporates fine-grained fitness assignment strategy which takes into account for each individual the number of individuals that dominate it and the number of individuals it dominates, 2) it uses a nearest neighbour density estimation technique which guides the search more efficiently, and 3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions. The pseudo code of SPEA2, as reported by Coello Coello et al. (2002), is:

```
Initialize population P
Create empty external set E
For i = 1 to Genmax
     Compute fitness of each individual in P and E
     Copy all non-dominated individuals in P and E to E
     Use the truncation operator to remove elements from
     E when the capacity of the files has been exceeded
     If the capacity of E has not been exceeded then use
     dominated individuals in P to fill E
     Perform binary tournament selection with replacement
     to fill the mating pool
     Apply crossover and mutation to the mating pool
End Loop
```

*Figure 7. SPEA2 pseudo code*

### 3.5.5 Multi-objective messy genetic algorithms (MOMGA)

This algorithm is proposed by Van Veldhuizen and Lamont (2000) as an attempt to extend the messy GA (Deb, 1991) to solve multi-objective optimization problems. MOMGA consists of three phases: 1) Initialization Phase where building blocks of a certain specified size are constructed through a deterministic procedure known as partially enumerative initialization, 2) Primordial Phase where tournament selection is performed and the size of population is reduced if necessary, 3) Juxtapositional Phase where the messy GA proceeds by building up the population through the use of the cut and splice recombination operator. A revised version called MOMGAII has been proposed by Zydallis et al. (2001) where the fast-messy GA (Goldberg et al., 1993) was extended. This consists also of three phases:

– Initialization Phase,

– Building Block Filtering Phase,

– Juxtaposition Phase.

The main differences are in the first phase where a probabilistic complete initialization is used to create a controlled number of building block clones of a specified size, and in the second phase where the number of building blocks is reduced by filtering and the best blocks are stored (elitism). The filtering process is accomplished through a random deletion of bits alternated with a tournament selection between blocks that have been found to generate “good” solutions. The pseudo code of MOMGAII, as reported by Coello Coello et al. (2002), is:

```
For n = 1 to k
     Perform probabilistically complete initialization
     Evaluate each pop member's fitness (w.r.t. k Templates)
     // Building Block Filtering Phase(BBF)
     for i = 1 to maximum number of BBF generations
          If(BBF required based off of input Schedule)
                 Then perform building block filtering
           Else
                  Perform tournament thresholding selection
           Endif
           End Loop
     // Juxtaposition Phase
     For i= 1 to maximum number of Juxt. generations
      Cut-and-splice
      Evaluate each pop member's fitness (w.r.t. k Templates)
      Perform tournament thresholding selection
      and fitness sharing
      P_known(t) = P_current(t) U P_known(t-1)
     End Loop
     Update k competitive Templates
            (using best value known in each objective)
End Loop
```

*Figure 8. MOMGAII pseudo code*

**Strengths and weaknesses:** Although messy GAs are very powerful, their main disadvantages are related to the exponential growth of their population when the size of the building blocks grows (Mitchell, 1996). The introduction of the messy GA was a good alternative to reduce this difficulty but does not eliminate it completely (Coello Coello et al., 2002).

### 3.5.6 Pareto-based selection

These approaches incorporate Pareto ranking as proposed by Goldberg (1989) but the diversity is maintained by mechanisms that are different from niching, crowding or fitness sharing. An example of these MOEAs is the Thermodynamically Genetic Algorithm (TDGA) proposed by Kita et al. (1996). This algorithm uses Pareto ranking in combination with the principle of minimal free energy used in simulated annealing. The fitness to be minimized is:

$$\text{fitness} = \langle E \rangle - HT \tag{12}$$

where $\langle E \rangle$ is the mean energy of the system, H the entropy and T the temperature. The diversity of the population is controlled by adjusting T according to a certain variation law (as in simulated annealing). The authors argue that this parameter is less sensitive to the population size and to the

dimension of the feasible region than the traditional sharing factor (Kita et al., 1996; Mori et al., 1995, Tamaki et al., 1996).

Another algorithm proposed by Osyczka and Kundu (1995, 1996) called "distance method" is based on the contact theorem (one of the main theorems in multi-objective optimization (Lin, 1976)) to determine relative distances of a solution with respect to the Pareto set.

**Strengths and weaknesses:** These approaches do not require any explicit sharing function, but remain highly sensitive to other parameters introduced to maintain diversity. For example, in the distance method, the penalty factor used to incorporate the constraints into each objective and the so-called starting distance, introduced in the fitness, equivalent to some scaling factor used to compare relative quality among solutions, have a noticeable effect on the algorithm's performance. If these two parameters values are not chosen properly, too much selection pressure may be generated or the GA may jump back and forth between feasible and non feasible regions, producing too many dominated solutions and consequently losing portions of the Pareto fronts (Coello Coello, 1999).

### 3.5.7 Pareto-demes based selection

Several approaches exist in the literature where the evolutionary algorithms are applied to subpopulations called demes (island model) that evolve independently. Each deme is interconnected by some defined topology or geographic structure used for communication; these communication channels are normally used for the occasional migration of individuals between demes. The island model can be executed on a sequential processor but its power is effective when using multiple processors. Pareto ranking is used to distribute these subsets within some sort of geographical structure. This scheme aimed to distribute the effort of checking for nondominance within each (presumably small) subpopulation of a (most likely parallelized) MOEA. An additional mechanism has to be used to determine dominance with respect to the entire population. Since only non-dominated individuals participate in this mechanism, the procedure is more efficient than using the entire population of a traditional (sequential) MOEA (Rowe et al., 1996).

**Strengths and weaknesses**: These techniques require parallel MOEA to exploit their power but parallelism introduces additional problems such as the cost of communication topology. For a more detailed discussion on parallel MOEAs the reader is referred to the book of Coello Coello et al. (2002).

### 3.5.8 Pareto-elitist-based selection

The use of elitism in the context of evolutionary multi-objective optimization has been reported since the mid 1990s. Elitism selection refers to retaining the best k individuals (generally $1 \leq k \leq N$) from the current generation and carrying them forward to the next generation without any transformation. Unlike single-objective GA optimization, the importance of elitism in MOEAs has not been yet proved (Laumanns et al, 2001). The main idea in elitist selection is to retain the highest ranked individuals in the population for the next generation and then create the rest using some other technique.

Although, most modern MOEAs (such as SPEA, NSGAII, PAES (Knowles and Corne, 2000b), MOMGA) use elitism, they do not use it as described above. The techniques using Pareto-elitist-based selection, and categorized in so far as done by Coello Coello et al. (2002), are:

– **Pareto Archived Evolution Strategy (PAES)**: This algorithm, proposed by Knowles and Corne (2000a) starts with one randomly initialized solution and then, one candidate is generated by means of mutation at each iteration. It is a (1+1) evolution strategy (i.e. a single parent that generates a single offspring). An external archive (of limited size) is maintained to collect non-dominated solutions. This archive is used as a reference set against which each mutated individual is being compared. An adaptive grid that divides the objective space is used to evaluate how many solution lies in each region. This is called the crowding procedure. Analogously to the NPGA, the candidate solution is discarded if it is dominated by the current solution or any other solution in the external archive. The candidate solution is added to the archive and becomes the current solution if it dominates the current solution. If none of them dominates the other, the selection is made on the crowding mechanism: the solution in the less crowded region is selected. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space). Furthermore, the procedure has a lower computational complexity than traditional niching methods (Knowles and Corne, 2000b). $(1+\lambda)$ and $(\mu+\lambda)$ variations of PAES were proposed also. However, the use of a population adds an important computational overhead (Knowles and Corne, 2000b). There is also a memetic version of PAES, M-PAES (Knowles and Corne, 2000a). This memetic algorithm incorporates a population and a crossover operator but uses the same selection mechanism of PAES. Two archives are used, one is the global archive of non-dominated solutions and another serves as the comparison set in the local search phase. The second archive is emptied after each local search and filled with solutions from global archive. The authors reported that this memetic version outperforms the original version on tests instances of multi-objective knapsack problems.

– **Pareto Envelope-based Selection Algorithm (PESA)**: PESA was proposed by Corne et al. (2000). It uses a small internal population and a larger external (or secondary) population. The same mechanism as in PAES is used to maintain the diversity. The crowding measure is used for the selection and to introduce a solution in the external population. The pseudo code of this algorithm, as reported by Coello Coello et al. (2002), is:

```
Generate a random (internal) population PI
Evaluate each member of PI
Initialize the external population PE to the empty set
     While termination criterion has not been met
     Incorporate non dominated individuals from PI into PE
     Delete the current contents of PI
     Repeat
            With probability Pc, select two parents from PE
      /*Pc = Probability of crossover*/
      Produce a single child via crossover
      Mutate the child created in the previous step
      With probability (1-Pc) select one parent
      Mutate the selected parent to produce child
     Until the population PI is filled
     End while
Return the members of PE as the result
```

*Figure 9. PESA pseudo code*

A revised version of this algorithm called PESA II is proposed (Corne et al., 2001) where the selection procedure is modified. The procedure consists of selecting, using any of traditional selection technique (Goldberg and Deb, 1991), a hyperbox and then randomly select an individual within such a hyperbox. The main motivation of this approach is to reduce the computational costs associated with MOEAs based on Pareto ranking.

– **The Micro-Genetic Algorithm for Multi-objective Optimization**: This approach, introduced by Coello Coello and Toscano Pulido (2001a, 2001b), works with a small population and a reinitialization process. The algorithm is illustrated on the pseudo code as reported by Coello Coello et al. (2002):

```
Generate randomly initial population
Divide this population into replaceable RP and non replaceable NRP
populations
For i= 1 to genmamx
Fill the population memory with RP and NRP
/*  micro-GA cycle*/
     Fill initial population of the micro-GA cycle from both RP and
     NRP      (individuals from RP and NRP are selected with a
     certain probability)
     While nominal  convergence criterion has not been met
          Selection
          Crossover
          Mutation
          Elitism
          New population RP
     End while
/* end of micro-GA cycle*/
Filter the new population RP
External memory (updated archive of non dominated individuals)
End Loop
```

*Figure 10. Micro-Genetic Algorithm for Multi-objective Optimization pseudo-code*

Initially, a random population is created to feed a *memory population* which is divided into two parts: a replaceable and a non replaceable portion. The non replaceable portion never changes during the entire run and provide the required diversity. The replaceable portion changes after each cycle in the micro-GA. During each cycle the micro-GA undergoes conventional genetic operators to evolve the population. This algorithm uses then three forms of elitism: 1) retain the non-dominated solutions found within the internal cycle of the micro-GA (updated archive), 2) use a replaceable memory whose contents is partially "refreshed" at a certain intervals (initial population entering the micro-GA cycle), 3) replace the population of the micro-GA by the best solutions found after a full internal cycle of the micro-GA (the filter step).

**Strengths and weaknesses:** The main criticism towards Pareto elitist approaches is that they may not retain diverse enough populations to reach a $PF_{known}$ truly representative of $PF_{true}$. This is because, the elitism mechanism retains more and more $P_{current}(t)$ in the generational population, so the remaining solutions may not provide enough diversity for effective exploration. This can induce large pressure selection and premature convergence (Coello Coello et al., 2002). Therefore, the number of non-dominated solutions retained for the next generation needs to be chosen carefully. PAES, for example, does not perform well in Pareto fronts that are disconnected because it does not keep in the external file the extremes of the objective function space (Coello Coello and Toscano Pulido, 2001b; Zitzler et al., 2001b). The use of an efficient approach to maintain diversity is crucial to make this technique effective (Coello Coello et al., 2002).

### 3.5.9 Hybrid Selection techniques

In this approach, the population capacities are exploited by using two or more cooperative search techniques. The selection procedure is then not necessarily identical for each evaluated solution and change after a certain number of generations. The choice of a technique is decided by an uncertainty management technique such as fuzzy logic (Voget and Kolonko, 1998; Zhou and Gen, 1999; Cochran et al., 2000; Gandibleux et al., 2001).

## 3.6 MOEA complexity

The main process that dominates EA execution time is the solution evaluation. MOEAs algorithmic complexity depends on the number of fitness evaluations, solutions comparisons and additional calculations that could be required. EVOPs complexity is not considered here. This complexity is not found in simple genetic algorithm (SGA). After fitness evaluation in an SGA, resultant values are stored in memory and no further computation is (normally) required as far as fitness is concerned. In an MOEA, these stored values are compared and/or combined which adds algorithmic complexity. As a reference to the complexity of the various MOEA techniques presented above, we summarize the "worst-case" reported by Coello Coello et al. (2002) in Table 2, with the following notation:

– $T_f$: solution evaluation time (assumed the same for each objective),

– G: number of generation,

– Z: number of fitness function

– m: number of solutions per processor (Pareto demes case)

– C: number of several runs in independent sampling techniques

The Pareto concepts are the most computation complexity due to the $O(N^2)$ cost of determining which solutions are Pareto optimal. The storage requirement is problem dependent and mandated by specific structures used.

## 3.7 MOEA citations

Coello Coello et al. (2002), have catalogued MOEAs citations in three graphs where the number of citations per year, by techniques, and by type, are reported. The most interesting, in our opinion, is the second graph, as it compares the popularity of the different approaches presented in the present MOEAs review. The number of publications during this past twenty years (up to 2001) are reported for each technique and summarized on Table 3. These data highlight the popularity of *a posteriori* techniques as their citations are almost four times more important than *a priori* and progressive categories combined. This gives rise to the following questioning (Coello Coello et al. (2002)): Does this imply a willingness by DMs to select solutions from (possibly) unbiased searches? Or is that DMs are unwilling (or unable) to assign priorities to objectives without further information? Making decision *a posteriori* may be worthy and less expensive in the long run than making decisions without the additional knowledge gained through initial or interactive search. Among the *a posteriori* techniques, the Pareto approaches are the most popular.

*Table 2. MOEA solution technique complexity (Coello Coello et al., 2002)*

| MOEA technique | Computational Complexity |
|---|---|
| SGA | $G\ T_f N$ |
| Lexicographic | $G\ T_f Z\ N + G\ Z\ N^2 - G\ Z\ N$ |
| Linear Combination | $G\ T_f Z\ N + G\ Z\ N - G\ N$ |
| Multiplicative | $G\ T_f Z\ N + G\ Z\ N - G\ N$ |
| Target Vector | $G\ T_f Z\ N + G\ Z^2 + 2G\ Z$ |
| Minmax | $G\ T_f Z\ N + 3\ G\ Z\ N$ |
| Independent sampling | $C\ (G\ T_f Z\ N + G\ Z\ N - G\ N)$ |
| Criterion selection | $G\ T_f Z\ N + G\ N$ |
| Aggregation selection | $G\ T_f Z\ N + G\ Z\ N - N$ |
| Pareto rank | $G\ T_f Z\ N + G\ Z\ N^2 - G\ Z\ N$ |
| Pareto niche and share | $G\ T_f Z\ N + G\ Z\ N^2 - G\ Z\ N + N^2$ |
| Pareto demes | $G\ T_f Z\ N + G\ \frac{m^2 Z}{n^2} - G\frac{mZ}{n} + \frac{m}{n} T_{comm}$ |
| Pareto elitist | $G\ T_f Z\ N + G\ Z\ N^2 - G\ Z\ N$ |

Elsewhere, Coello Coello et al. (2002) noted that the number of papers comparing MOEAs is a "healthy sign of scepticism", in that researchers are seeking to compare proposed algorithms on a variety of problems. MOEA theoretical concerns are poorly represented in this diagram, indicating the necessity of further development in this field to (possibly) increase the effectiveness and efficiency of existing MOEAs.

*Table 3. Statistics on MOEAs Publications in Scientific Literature*

| Application category | Approximate Number of publications |
|---|---|
| Lexicographic | 6 |
| Linear Fitness Combination | 71 |
| Nonlinear Fitness combination | 34 |
| Progressive | 30 |
| Independent sampling | 10 |
| Criterion selection | 27 |
| Aggregating selection | 32 |
| Pareto Ranking | 90 |
| Pareto Rank- & Niche-Based Selection | 139 |
| Pareto Deme-Based Selection | 20 |
| Pareto Elitist-Based | 30 |
| Hybrid selection | 4 |
| Technique Comparisons | 84 |
| Theory and Reviews | 30 |

# 4 A genetic algorithm for a multi-objective course of action planning

Planning military courses of action (COAs) is a very complex and difficult activity. Planners should take into consideration environmental information, predictions, the end state targeted and resource constraints. Development of courses of action involves solving simultaneously planning and scheduling problems. In this section, a new approach based on genetic algorithms (GA) and multi-objective optimization is proposed to support resource-constrained courses of action development where both cardinal and ordinal objectives are considered. A vector of fitness evaluations is proposed to control the proportion of the infeasible solutions. Crossover and mutation operators are designed to diversify the search space and improve solutions (COAs) on all objectives from one generation to another. In the replacement strategy, a selection procedure, based on the dominance concept and a multi-criteria-filtering method, is proposed. Such a strategy is applied when the population reaches a critical size. Different GA schemes are compared and their strengths and weaknesses are discussed using some performance metrics from the literature.

## 4.1 Formulation of the problem

Multi-objective courses of action could be characterised by a set of tasks, a set of resources, precedence relationships, resource availability constraints and global performance functions (criteria) *F*. The problem formulation states as follows:

$$\text{Optimise } f_z,\ z = 1,\ldots, Z \tag{13}$$

$$\text{s.t.} \quad t \in D \tag{14}$$

$$\text{s.t.} \quad R \in C \tag{15}$$

with a vector of tasks $t = \{ t_1,t_2,\ldots.t_n\}$ having the following attributes for each task $t_i$:

- Starting and ending time [td(i),tf(i)] considered as integer variables. The earliest starting and latest ending time, respectively $[\tau_s(i),\tau_e(i)]$, are used to generate different initial solutions.
- Starting and ending time [td(i),tf(i)] considered as integer variables. The earliest starting and latest ending time, respectively $[\tau_s(i),\tau_e(i)]$, are used to generate different initial solutions.
- A localization spatial coordinate (x,y,z).
- Type and quantity of resources required, represented by a set **R** composed of renewable and non-renewable resources available in limited quantities. $R_k(t_i) = \{r_{1i}, r_{2i},\ldots.r_{mi}\}$ is the $k^{th}$ set (or combination) of resources required to accomplish the task $t_i$.
- Set of predecessors {PR} characterized by the tasks which temporally and/or spatially precede $t_i$.

and resources with the following attributes:

- Starting and ending time of availability [$t_{rs}(k)$, $t_{re}(k)$] (resource's timetable).
- Quantity available during this interval of time.
- Localization of the resources (depot) (x, y, z).
- Type of resource.
- Other specific characteristics such as in-use cost, mean speed (for mobile resources), reliability, etc.

Equation (14) represents the constraints that ensure that each task is processed once in its time interval and precedence conditions are fulfilled (feasible tasks). Constraints in equation (15) express the resource constraints (*e.g.*, availability).

The mGA is used to find different task-resource combination networks where all activities are completed, the resource and precedence constraints satisfied, and the best compromise between criteria reached. The optimization is carried out using a randomly initialised population for a low size problem (COAs with less than 10 actions). For large size problems (COAs with more than 10 actions), a heuristic based on the network approach and CPlex is used (Urli et al., 2003) to obtain initial population. Crossover and mutation are used for the exploration and exploitation of the search space. The principle of the variable neighbourhood search (VNS) method is used during the optimization. This is achieved by introducing three binary (crossover, mutation) and one unary (mutation) operators in order to explore a great number of neighbourhoods. Unlike classical GA procedures, the population size is not kept constant from generation to generation but increases by reproduction and mutation until it reaches a critical value determined empirically. A replacement strategy is then applied to select the best candidates for the next generation (survivals). The selection is achieved based on dominance and/or a 1$^{st}$ order multi-criteria filtering procedure (MFP).

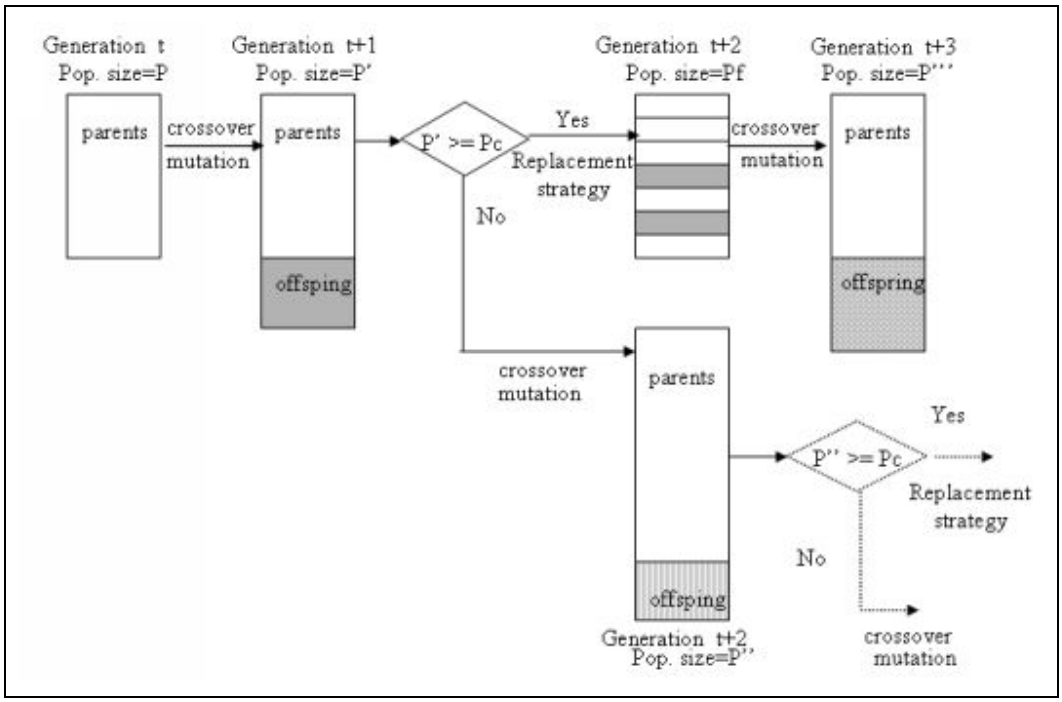


*Figure 11. Evolution of the population from generation to generation*

### 4.1.1 Encoding variables

In the genetic algorithm, a COA is encoded in the following form:

$$x(i) = \left[ (t_1, R_{1,1}), (t_2, R_{2,4}), ....., (t_n, R_{n,1}) \right] \quad \text{for i =1,…, popsize} \tag{16}$$

where popsize is the population size, $t_j$ is the $j^{th}$ task to be scheduled and $R_{j,k}= R_k(t_j)$ is one of the sets of resources available to accomplish this task.

### 4.1.2 Evaluation of a solution

As it is difficult to apply genetic algorithms to a multi-objective, constrained problem, one of the options proposed in the literature is to integrate the constraints into the objective functions using penalty functions. Fitness (or evaluation) functions obtained in this way are penalized according to the distance of infeasible solutions from the limits of feasible space. However, the aggregation of penalty functions to the objectives in a highly constrained problem may lead to a large proportion of unfeasible solutions and there is no control of this number. In this work, the constraints are considered as functions to be optimised like the objectives and the solutions generated will be retained regarding both their objectives and constraint violation. Thus, to each solution, a fitness vector is assigned.

### 4.1.3 Constraints functions

**Resource availability**: Each resource, $r_{kj}$, used by task $t_j$ must be available, in the desired quantity, during the period $[t_{rs}(j), t_{re}(j)]$. A penalty coefficient DP(j) associated to each task is introduced and defined by:

$$DP(j) = \begin{cases} 1, \text{if all the resources to accomplish the task } t_j \text{ are available} \\ 0, \text{otherwise} \end{cases} \quad (17)$$

For each solution, it follows:

$$\text{fitn}_1(i) = \sum_{j=1}^{n} PP(j) \quad (18)$$

This function is to be maximised (arg [max $\text{fitn}_1(i)$] = n) to force the search into the feasible region.

**Precedence**: Some tasks have successors and predecessors. Other tasks are free. For each task j, if all its predecessors have been correctly scheduled, we set tPP(j) =1 otherwise tPP(j) = 0. The precedence penalty is defined as:

$$\text{fitn}_2(i) = PP(i) = \sum_{j=1}^{n} tPP(j) \quad (19)$$

$\text{fitn}_2(i)$ is to be maximised.

### 4.1.4 Objective functions

The objectives considered here could be cardinal (quantitative) such as cost, reliability, make-span, or ordinal (qualitative) such as impact of a COA. For a uniform optimization (maximisation), the objectives to be minimised are reformulated.

## 4.2 Genetic operators

### 4.2.1 Crossover (or recombination)

Selection of two parents for the crossover is done using the roulette wheel selection. Two candidates are selected using the following procedure repeated popsize times:

– Choose randomly two fitness components $\text{fitn}_k$ and $\text{fitn}_l$ (objectives or constraints)

– Evaluate selection probabilities of the population according to these two fitness components:

$$ps_k(i) = \frac{fitn_k(i)}{\sum_{i=1}^{popsize} fitn_k(i)} \tag{20}$$

$$ps_l(i) = \frac{fitn_l(i)}{\sum_{i=1}^{popsize} fitn_l(i)} \tag{21}$$

- Select randomly one candidate per fitness according to its selection probability.
- This selection procedure, favouring one criterion per candidate, aims to produce offspring characterised by the best features present in their parents, and ultimately induces higher population diversity.

Two crossover procedures are proposed, to be used alternatively, in order to explore a greater number of search spaces. The first one is the uniform crossover operator (Syswerda, 1989) which has been shown to be superior to traditional crossover strategies for combinatorial problems. When two chromosomes are selected for crossover, a random mask is generated and their genes are exchanged according to the mask. This mask is simply a binary string with the same length as a COA vector. The parity of each bit determines which genes will be exchanged. The gene in this procedure is represented by an action ($t_j$,$R_{j,k}$) (see Figure 12).

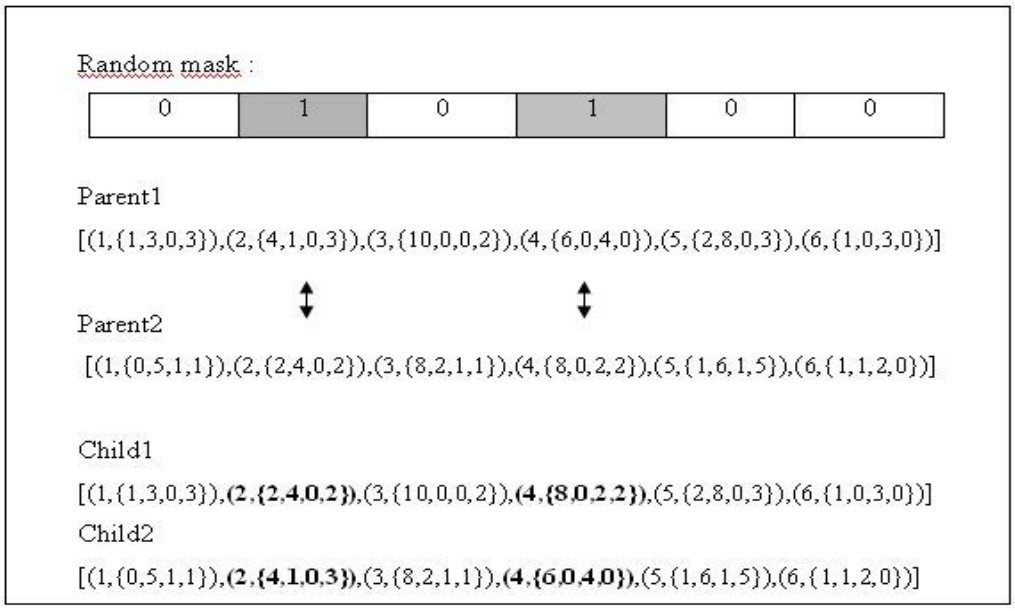


*Figure 12. Illustration of uniform crossover operation*

The second operator is the partial mapped crossover (PMX) proposed by Goldberg and Lingle (1985) and is an extension of two-point crossover to permutation representation. Select two positions randomly:

$$\begin{aligned}&\text{Parent1} \quad [(t_1,\{R_1\}),\ (t_2,\{R_2\}),\ (t_3,\{R_3\}),\ (t_4,\{R_4\}),\ (t_5,\{R_5\}),\ (t_6,\{R_6\})]\\&\text{Parent2} \quad [(t'_1,\{R'_1\}),(t'_2,\{R'_2\}),(t'_4,\{R'_4\}),(t'_5,\{R'_5\}),(t'_3,\{R'_3\}),(t'_6,\{R'_6\})]\end{aligned} \tag{22}$$

Exchange a substring between parents:

$$\begin{aligned}&\text{Proto-child1} \quad [(t_1,\{R_1\}),\ (t_2,\{R_2\}),\ (t'_4,\{R'_4\}),(t'_5,\{R'_5\}),(t_5,\{R_5\}),\ (t_6,\{R_6\})]\\&\text{Proto-child2} \quad [(t'_1,\{R'_1\}),(t'_2,\{R'_2\}),\ (t_3,\{R_3\}),\ (t_4,\{R_4\}),\ (t'_3,\{R'_3\}),(t'_6,\{R'_6\})]\end{aligned} \tag{23}$$

Determine mapping relationship: in proto-child1, the action 3 $(t_3,\{R_3\})$ is missing whereas the action 5 $(t_5,\{R_5\})$ is missing in proto-child2. Legalization is done by transferring action 3 in child1 and action 5 in child 2.

The offspring are then:

$$\begin{aligned}&\text{Child1} \quad [(t_1,\{R_1\}),\ (t_2,\{R_2\}),\ (t'_4,\{R'_4\}),(t'_5,\{R'_5\}),(t'_3,\{R'_3\}),(t_6,\{R_6\})]\\&\text{Child2} \quad [(t'_1,\{R'_1\}),(t'_2,\{R'_2\}),\ (t_3,\{R_3\}),\ (t_4,\{R_4\}),\ (t_5,\{R_5\}),(t'_6,\{R'_6\})]\end{aligned} \tag{24}$$

The repairing procedure is used, in both operators (uniform and PMX crossover), to resolve illegitimacy of the offspring if some activities are missing or duplicated. This procedure is achieved by simply transferring these activities from one child to the other one.

### 4.2.2 Mutation

Mutation is applied randomly on the population and the probability of mutation is inversely proportional to the population size as recommended by De Jong (1975).

The two operators used for the mutation consist of:

– Exchanging, with a probability $p_\alpha$, a randomly selected combination of resources between two COAs. The offspring, which received the combination with the best criterion, is retained.

– Switching the quantity of two resources in a combination of resources associated to a randomly selected task j. This option is used every four generations.

The probability of mutation $p_\alpha$ of a set of resources $R_{i,j}$ in a chromosome is based on a criterion related to the resource characteristics such as cost or reliability. In this way, only the best resource combination is exchanged, in regard to the selected criteria. These probabilities are used alternatively from one generation to another, to produce COAs with improved combinations of resources. Here two examples follow on how these probabilities are expressed:

Cost: $$p_{cost}(R_{i,j}) = \frac{cost\,max - cost(j) + \varepsilon}{\sum_{j}(cost\,max - cost(j) + \varepsilon)} \quad (25)$$

Reliability: $$p_{rel}(R_{i,j}) = \frac{Rel(j)}{\sum_{j} Rel(j)} \quad (26)$$

ε positive constant less than 1 chosen to ensure that $0<p_{cost}(R_{i,j})<1$. cost(j) is the total cost of resources combined in the set $R_{i,j,}$ $cost_{max}$ is the highest set's cost used in the project. Re(j) is the global reliability of $R_{i,j}$ calculated on the reliability of all resources used in this set.

### 4.2.3 Replacement strategy

The solutions generated from the crossover and the mutation operations are evaluated. When the population size attains or exceeds a critical value $P_c$, $P_f$ individuals are selected, based on their fitness vector, among parents and offspring using a replacement procedure. Otherwise, new offspring are generated (see Figure 11). Three strategies are tested for the replacement:

– The multi-criteria filtering procedure (MFP), proposed by Guitouni et al. (2001), returns (a user-defined) $P_f$ non-dominated diversified individuals. The MFP is based on multi-criteria dynamic conjunctive and disjunctive procedures. The retained solutions are characterised by at least one best-scored objective or by all objectives achieving minimal threshold values

– The non-dominated sorting approach (NDS), as proposed by Srinivas and Deb (1995), but used only to rank the best $P_f$ individuals

– The mixed approach: NDS+MFP procedure.

**Strategy 1 with MFP method**: The filtering procedure is performed in two steps using a disjunctive procedure then a conjunctive procedure. Let A be the set of parents and offspring to be filtered and $\hat{A}$ the set of individuals retained for the replacement. Let Card(A) = $P_c$ and Card($\hat{A}$) = $P_f$. Let $\underline{e}_0$ be a threshold vector representing the lower limits (lower thresholds) of the objectives imposed to the selection: $\underline{e}_0 = (e_{01}, e_{02},\ldots, e_{0Z})$. Let $\underline{e}_1$ be a threshold vector representing the upper limits (higher thresholds) of the objectives imposed to the selection: $\underline{e}_1=(e_{01}, e_{02},\ldots, e_{0Z})$. $S_1$: $\{fitn_{1min}, fitn_{2min},\ldots, fitn_{Zmin}\}$ the set of anti-ideal points of A, $fitn_{z,min} = \min_{i=1,Pc} \{fitn_z(i)\}$, z = 1… Z, $S_2$: $\{ fitn_{1max}, fitn_{2max},\ldots, fitn_{Zmax}\}$ the set of ideal points of A, $fitn_{z,max} = \max_{i=1,Pc} \{fitn_z(i)\}$, z =1… Z.

**Disjunctive procedure**: This procedure selects individuals characterised by at least one objective having a maximal value. Let $\hat{A}_d$ be the set of individuals selected by the disjunctive method. The following steps could describe it:

– $\hat{A}_d = \emptyset$

– Compute $e_{1z}$ for each objective ($e_{1z} = S_{2z} = fitn_{z,max}$, z =1,…Z)

– Select individuals i that $\exists z \in \{1,\ldots,Z\}$, $fitn_z(i) \geq e_{1z}$, i = 1…$P_c$

– Add these individuals in $\hat{A}_d$.

**Conjunctive procedure**: Candidates characterised by objectives less than the lower thresholds are discarded. The thresholds are automatically computed using a dichotomist method between $S_1$ and $S_2$. The procedure has the following steps:

– Compute the threshold value $e_{0z}$ for each objective

– Select individual i so that $\forall$ z, $fitn_z(i) \geq e_{0z}$, i = 1…$P_c$

– Add this individual in the set $\hat{A}_d$ until Card($\hat{A}_d$) = Card($\hat{A}$).

This multi-objective filtering procedure allows the selection of a diversified population of individuals characterised by the fittest objective or all objectives higher than a threshold value.

**Strategy 2 with NDS method:** The solutions are ranked based on non-domination. All non-dominated individuals in the current population are placed at the top of a list and assigned a rank of 1. These solutions are removed from the remaining population and the next set of non-dominated solutions is identified and assigned rank 2. The process is repeated until the entire population is ranked. The top $P_f$ individuals in the list are then selected for the next generation.

**Strategy 3 with NDS+MFP method**: This procedure combines the features of the two previous ones. First, the population is ranked using the NDS method. The set identified with rank 1, *i.e.*, the non-dominated solutions, is selected. Let $A_1$ be this set.

– If card($A_1$) = $P_f$ , then these solutions are individuals of the new generation

– If card($A_1$) > $P_f$, use the MFP method to return $P_f$ solutions

– If card($A_1$) < $P_f$, select the set of solutions identified with rank 2. Let $A_2$ be this set

– If card ($A_1 \cup A_2$) >$P_f$, use the MFP method to return $P_f$ solutions

– If card ($A_1 \cup A_2$) < $P_f$,, select the set of solutions identified with rank 3, and repeat the process until $P_f$ solutions are selected.

This strategy differs from NDS because it selects the solutions based on their evaluations besides being non-dominated.

The efficiency of these replacement strategies is examined in next chapters.

## 4.3 Algorithm of the proposed procedure

The proposed procedure is implemented as follows:

**Inputs**: Genmax: maximum number of generations (stopping criterion)

N: population size

$p_c$: probability of crossover

$p_m$: probability of mutation = 1/N

$P_f$ = size of the population returned by the filtration procedure

Pc = size of the population to be filtered

**Output**: ParetoSet P*: Archived non-dominated solutions set

**Step 1:** Initialisation: read initial population Po and evaluate it (calculate the fitness vector of each solution). Create the empty Pareto set P*o=∅. Set gen = 0.

**Step 2:** Store statistical information (diversity, fitness max components, fitness min components, fitness mean components, % of feasible solution, etc)

**Step 3:** Probability of selection for crossover

**Step 4:** Apply crossover and mutation to generate offspring and evaluate them

if (gen modulo 3 = 0) then use PMX crossover else use the uniform crossover

if (gen module 4 = 0) then use random switching of two resources inside a solution else use exchanging resource combination between two solutions

**Step 5:** Copy all solutions of P(gen) and offspring into P(gen)

**Step 6:** if the size of P(gen) >= Pc then apply filtering procedure (NDS, MFP, or NDS+MFP) which return $P_f$ best solutions. Copy these best solutions in P(gen+1) and increment gen (gen = gen+1)

**Step 7:** Archive and update P*(gen) : Copy the non-dominated solutions of P into P* and delete the dominated solutions in P*

**Step 8:** if stopping criterion is reached (gen =genmax) stop else go to Step 2.

*Figure 13: This is the caption for the figure shown above.*

# 5 Measurement of performance of multi-objective optimizers

A great number of publications are dedicated, this past decade, to the comparison of the performance of multi-objective approaches proposed in the literature. Different investigations to evaluate the performance of these algorithms have been proposed, taking into account the optimisers' stochastic characteristics (see for example: da Fonseca et al., 2001) or not (see for example Zitzler et al., 2003) and often assuming that the Pareto front is known. As pointed out by Bosman and Thierens (2003), comparing the performances of multi-objective evolutionary techniques is not an easy task. There are several criteria of finding a good approximation of the Pareto front and most current methods do not outperform each other but work better with regard to different performance indicators. In this section, we examine the different metrics proposed in the literature to evaluate the quality of the *approximations Pareto sets* and the limitations of this type of quality assessment. This work is based essentially on the following references: Coello Coello et al. (2002), Knowles and Corne (2002), and Zitzler et al. (2003).

## 5.1 Multi-objective optimization goals and approximation sets

Multi-objective optimization generates not a single solution but a set of non-dominated solutions. Generating the Pareto optimal set, $P_{true}$, may be computationally expensive and often infeasible, that is why metaheuristics such as evolutionary algorithms (EAs) are often used instead of exact methods. These optimization tools do not guarantee to identify $P_{true}$, but try to find a good approximation set $P_{known}$ of solutions whose objectives are not far from the Pareto front. Moreover, regardless the size of $P_{true}$, it is desirable to find an approximation set that contains a selection of solutions of $P_{true}$ such that these solutions are as diverse as possible. This introduces the concept of **proximity** and **diversity**, the two-sided goal of MOEAs. As reported by Bosman and Thierens (2003), there is a trade-off to make between a diversified approximation set (diversification) and an approximation set close to the Pareto front (proximity). Regarding the diversity, it depends on the mapping between the decision variable space and the objective space. Generally, MOEAs researchers have focused on generating $PF_{true}$, so the diversity of the solutions is in the objective space along the approximation set (Coello Coello, 1999).

## 5.2 Comparison methods

The notion of performance, when comparing two methods, includes both the quality of the results (outcome) and the computational requirements. Concerning the latter aspect, it is usually related to the complexity and the overall run-time on a particular computer. As to the quality aspect, it gets too complicated to compare the outcome (approximation sets) of two methods, because some solutions found by either method (in either set) may be dominated by solutions found by the other while others may be incomparable. Several studies, in the literature, addressed the problem of comparing approximations sets in a quantitative manner. Before examining these tools, we should define the relation of comparison between sets as stated by Zitzler et al. (2003) which is, in our opinion, the most rigorous analysis published so far.

Five relations are distinguished to compare two approximation sets A and B, based on dominance concept:

**A strictly dominates B** ($A \succ\succ B$) means that every solution $x_B \in B$ is **absolutely** dominated by at least one solution $x_A \in A$, (27)

**A dominates B** ($A \succ B$) means that every solution $x_B \in B$ is **strictly** dominated by at least one solution $x_A \in A$, (28)

**A is better than B** ($A \rhd B$) means that every solution $x_B \in B$ is **weakly dominated** by at least one solution $x_A \in A$ and $A \neq B$, (29)

**A weakly dominates B** ($A \succeq B$) means that every solution $x_B \in B$ **is weakly dominated** by at least one solution $x_A \in A$, (30)

A and B are **incomparable** (A || B) means that neither A dominated B nor B dominates A. (31)

The authors have denoted a natural ordering among the relations:

$$A \succ\succ B \Rightarrow A \succ B \Rightarrow A \rhd B \Rightarrow A \succeq B \quad (32)$$

$$A \succeq B \Rightarrow A \rhd B \vee A=B \quad (33)$$

They gave the definition of a quality indicator as:

– **Quality indicator definition**: An m-ary quality indicator I is a function I: $\Omega^m \rightarrow IR$, which assigns each vector $(A_1,A_2,...,A_m)$ of m approximation sets a real value $I(A_1,A_2,...,A_m)$.

This definition means that quality measures map approximations sets to the set of real numbers. Moreover, an interpretation of this quality number is necessary. Thus, the authors proposed an interpretation function E which maps vectors of real numbers to Booleans. For example for the quality number "generational distance", $I_{GD}$, they defined $E(I_{GD}(A), I_{GD}(B)) := (I_{GD}(A) = 0 \wedge, I_{GD}(B)>0)$, i.e., E is true if and only if $I_{GD}(A) = 0$ and at the same time $I_{GD}(B) > 0$. Such a combination of one or more quality indicators and an interpretation function is called a **comparison method**. The authors focused their definition on methods that: 1) consider two approximation sets only, 2) use either only unary or only binary indicators.

– ***Comparison method* definition**: Let A, B, $\in \Omega$ (the set of all approximation sets) be two approximation sets, $I = (I_1, I_2, ...I_k)$ a combination of quality indicators, and E: $IR^k \times IR^k \rightarrow$ {false,true} an interpretation function which maps two real vectors of length k to a Boolean value. If all indicators in I are unary, the **comparison method $C_{I,E}$** defined by I and E is a function of the form:

$$C_{I,E}(A,B) = E(I(A), I(B)) \quad (34)$$

where $I(A) = ( I_1(A), I_2(A),...,I_k(A))$. If I contains only binary indicators, the comparison method $C_{I,E}$ is defined as:

$$C_{I,E}(A,B) = E(I(A,B), I(B,A)) \quad (35)$$

where $I(A,B) = ( I_1(A,B), I_2(A,B),...,I_k(A,B))$.

This comparison method is characterized by two properties: the **compatibility** and the **completeness** which are defined as:

– ***Compatibility and* completeness**: Let ► be an arbitrary binary relation on approximation sets. The comparison method $C_{I,E}$ is denoted as ►c*ompatible*

if either of any A,B , $C_{I,E}(A , B) \Rightarrow A \blacktriangleright B$

or (36)

for any $A,B \in \Omega$ , $C_{I,E}(A , B) \Rightarrow B \blacktriangleright A$

The comparison method $C_{I,E}$ is denoted as ►c*omplete*

if either of any $A,B \in \Omega\, A \blacktriangleright B \Rightarrow C_{I,E}(A, B)$

or (37)

for any $A, B \in \Omega\ B \blacktriangleright A \Rightarrow C_{I,E}(A, B)$

The goal of a comparative study is to indicate the differences in performance between algorithms. Then, it is compliant with the statement "algorithm A outperforms algorithm B" should imply that the outcome (approximation set A) of algorithm A is better than the outcome (approximation set B) of algorithm B i.e., $A \triangleright B$ (the authors assume that a single run of the algorithms is performed). Thus, a good comparison method should be compatible and complete, i.e.,

$$C_{I,E}(A , B) \Leftrightarrow A \blacktriangleright B \text{ or } C_{I,E}(A , B) \Leftrightarrow B \blacktriangleright A \quad (38)$$

## 5.3 Performance assessment based on unary quality indicators

Here follows the most important unary indicators reported in the literature. Some of them measure the deviation from the Pareto front of the approximation set (the proximity) and others measure the diversity of this approximation set. Many of these metrics may also be used to evaluate performance of generational population during the execution of the algorithm. They are

attractive as they are able to assign quality values to an approximation set independently of other sets under consideration. However, they have some important limitations as shown below.

### 5.3.1 Error Ratio, $I_{ER}$

This metric, proposed by Van Veldhuizen (1999), assume that $PF_{true}$ is known. It is defined as:

$$I_{ER} = \frac{\sum_{i=1}^{n} e_i}{n} \tag{39}$$

where $e_i = 0$ if a solution $x_i$ of the approximation set $PF_{know}$ is a member of $PF_{true}$ otherwise $e_i = 1$. For example, $I_{ER} = 0$ indicates every solution reported by the algorithm in $PF_{known}$ is actually in $PF_{true}$; E = 1 indicates that none are.

Ulungu et al. (1999) proposed a similar metric calculating the ratio of the number of objective vectors in the approximation set to the number of all Pareto-optimal objective vectors. The authors proposed to use the supported efficient solution set when $PF_{true}$ is not available. To obtain this supported efficient set, they suggested solving the problem using the classical aggregative method with different sets of weights. Obviously, this is not appropriate when the search space is not convex.

A similar metric (Zitzler and Thiele, 1998a; Zitzler and Thiele, 1998b; Zitzler and Thiele, 1999) measures the percentage of solutions in a set (e.g. $Pk_{nown}$ or $PF_{known}$) dominated by another solution set's members (e.g. $P_{true}$ or $PF_{true}$). This type of indicators characterizes the proximity of an approximation set.

**Strength and weaknesses:** This metric is easy to implement and scaling independent (no normalisation of decisions variables or objectives is needed). However it necessitates a reference set ($P_{true}$ or $P_{reference}$). Moreover, this metric is not complete with respect to any dominance relation and is $\not\trianglerighteq$-compatible, meaning that we cannot necessarily state, according to the definition in section 5.2, that an algorithm outperforms another (Knowles and Corne, 2002; Zitzler et al., 2003).

### 5.3.2 Two set coverage, $I_{CS}$

Zitzler et al. (2000) proposed a metric of comparison of two sets where one is included in the other. $I_{CS}$ is defined as the mapping of the order pair (A,B), $A,B \subseteq \Omega$ ($\Omega$ the feasible space) being two sets of decision variable vectors, to the interval [0,1] per equation:

$$I_{CS}(A, B) = \frac{\left|\{\underline{x}_B \in B; \exists \underline{x}_A \in A : \underline{x}_A \succeq \underline{x}_B\}\right|}{|B|} \tag{40}$$

If all points in B are dominated or equal to solutions in A, then, by definition, $I_{CS}(A, B) = 1$. The opposite, $I_{CS}(A, B) = 0$ implies that none of the solutions in B are covered by the set A. In general, both $I_{CS}(A, B)$ and $I_{CS}(B, A)$ have to be considered because the intersection between the two sets is not empty. $I_{CS}$ characterizes the proximity of an approximation set and is a binary rather than a unary indicator.

**Strength and weaknesses:** The advantage of this metric is that it is easy to calculate and provides a relative comparison based on dominance numbers between two MOEAs. This is not a distance metric and does not require the knowledge of $P_{true}$ or $P_{reference}$. However, Knowles and Corne (2002) showed that the non-symmetric nature of this metric complicates the analysis of its compatibility with the $\rhd$ relation. If any pair of $I_{CS}$ metrics for two sets A and B has neither $I_{CS}(A,B) = 1$ nor $I_{CS}(B,A) =1$, this indicates that the two sets are incomparable according to the weak dominance relation but no further conclusion could be drawing. Nevertheless, this metric is $\succ\succ$ and $\succ$ compatible (Knowles and Corne, 2002) and compatible and complete with $\rhd$, $\succeq$, = and || relations (Zitzler et al. 2003)

### 5.3.3 Generational distance, $I_{GD}$

This metric (Van Veldhuizen, 1999) measures the convergence of an MOEA and evaluates how far $PF_{known}$ is from the nearest point of $PF_{true}$:

$$I_{GD} = \frac{1}{|\,PF_{known}\,|} \sum_{z^1 \in PF_{known}} \min_{z^o \in PF_{true}} \{d(z^o, z^1) \tag{41}$$

where $z^1$ is an objective vector member of $PF_{known}$, $z^o$ an objective vector member of $PF_{true}$ , and $d(z^o,z^1)$ the Euclidean distance between $z^o$ and $z^1$. Lower values represent better sets. This indicator characterizes the proximity of $PF_{known}$.

**Strength and weaknesses:** Zitzler et al. (2003) showed, by an example, that this metric does not provide sufficient information to conclude that $A \rhd B$, $A \not\rhd B$, etc. Moreover, it requires the knowledge of $P_{true}$ or $P_{reference}$ and introduces scaling issues that cannot be properly resolved without reference to additional preference information.

### 5.3.4 Maximum Pareto front error, $I_{ME}$

This is defined as follows (Van Veldhuizen, 1999):

$$I_{ME} = \max_j \left( \min_i \left| f_1^i(\underline{x}) - f_1^j(\underline{x}) \right|^\alpha + \left| f_2^i(\underline{x}) - f_2^j(\underline{x}) \right|^\alpha + \ldots + \left| f_Z^i(\underline{x}) - f_Z^j(\underline{x}) \right|^\alpha \right)^{1/\alpha} \tag{42}$$

where i =1,2,...,$n_1$ and j =1,2,...,$n_2$ are index vectors in $PF_{known}$ and $PF_{true}$, respectively. $I_{ME}$ measures the largest distance between any vector in $PF_{known}$ and the corresponding closest vector in $PF_{true}$ (the proximity).

**Strength and weaknesses:** Same as for the $I_{GD}$ indicator.

### 5.3.5 Distance from a reference set, $I_D$

Czyzak and Jaszkiewicz (1998) proposed the following metric:

$$I_D = \frac{1}{|R|} \sum_{r \in R} \min_{z \in A} \{d(r,z)\} \tag{43}$$

where A and R are the approximation and a reference sets, respectively, z vectors $\in$ A, r vectors $\in$ R, $d(r,z) = \max_k \{\gamma_k(r_k - z_k)\}$ and $\Gamma = (\gamma_1,\gamma_2,..,\gamma_Z)$ with $\gamma_k = 1/R_k$, k=1,..Z, with $R_k$ being the range of objective k in set R. $I_D$ measures the mean distance, over the points in a reference set, of the nearest point in an approximation set (the proximity).

**Strength and weaknesses:** This metric is cheap to compute and can differentiate between different level of complete out performance given an appropriate choice of reference set. Zitzler et al. (2003) showed that it is not $\succ$-complete, but $\rhd$-complete and $\not\rhd$-compatible. Hence, this metric can detect only if an approximation set is not worse than another. Moreover, it calculates weighted average where the reference points have equal weights. The score is then strongly dependent upon the distribution of points in the reference set and on the choice of $\Gamma$.

### 5.3.6 Distance from the Pareto optimal front, $I_{DP}$:

Bosman and Thierens (2003) proposed a metric similar to the abovementioned one, where $PF_{true}$ is considered instead of a reference set.

$$I_{DP} = \frac{1}{|PF_{true}|} \sum_{z^o \in PF_{true}} \min_{z^1 \in A} \{d(z^o,z^1)\} \tag{44}$$

The interesting point, advanced by the authors in using this metric, is the possibility of comparing the diversity-proximity tradeoff of approximation sets that are incomparable. If A and B are two incomparable approximations sets, and if $I_{DP}(A) < I_{DP}(B)$ then A is preferred to B.

**Strength and weaknesses:** This metric presents the same weaknesses as the precedent one ($I_D$) and a major drawback: in real applications, the true Pareto front is not known beforehand.

### 5.3.7 Coverage error, $I_{CE}$

Sayin (2000) proposed to evaluate the quality of a set A of solutions, generated from an MLPO (multiobjective linear problem optimization), by using three metrics: the coverage, the uniformity and the cardinality. From coverage they mean that all of the efficient solutions must be well represented. So A must contain points from every portion of $P_{true}$. This approximation set must be uniform, so it does not include redundancies or clusters because they do not contribute as much to the information being presented to the DM. The ideal approximation set will contain solutions that are equidistant according to a given distance metric. This approximation set must contain a reasonable number of solutions to facilitate the information processing of the DM. The definition of the coverage advanced by the author is:

Let be $\varepsilon > 0$ a real number. Let $A \subseteq P_{true}$ be a discrete set. A is called a $d_\varepsilon$-*representative* of $P_{true}$ if for any $\underline{x}^o \in P_{true}$, there exists $\underline{x}^1 \in A$ such that $d(\underline{x}^o, \underline{x}^1) \leq \varepsilon$.

Given this definition, the expression of the coverage error is:

$$I_{CE}(A) = \max_{\underline{x}^o \in P_{true}} \min_{\underline{x}^1 \in A} d(\underline{x}^o, \underline{x}^1) \tag{45}$$

The distance $d(\underline{x}^o, \underline{x}^1)$, called as the $l^p$ metric, has the following definition:

$$l^p(\underline{x}^o, \underline{x}^1) = \left( \left| x_1^o - x_1^1 \right|^p + \left| x_2^o - x_2^1 \right|^p + \ldots + \left| x_z^o - x_z^1 \right|^p \right)^{1/p} \quad \text{for } p = 1,2, \ldots$$

$$l^p(\underline{x}^o, \underline{x}^1) = \max_{i=1,\ldots,z} \left| x_i^o - x_i^1 \right| \qquad \text{for } p = \infty \tag{46}$$

This metric, is the same as the maximum Pareto front indicator, $I_{ME}$, proposed by Van Veldhuizen (1999). The uniformity indicator, $I_{MD}$, is presented in section 5.3.13.

**Strength and weaknesses:** Same as for the $I_{DP}$ indicator.

### 5.3.8 Overall non-dominated vector generation and ratio, $I_{ONGV}$, $I_{ONGVR}$

Most MOEAs add $P_{current}$ to $P_{known}$ each generation, possibly resulting in different cardinalities for $P_{known}$. This metric then measures the total number of non-dominated vectors found during MOEA execution and is defined as:

$$I_{ONVG} = | PF_{known} | \tag{47}$$

Schott (1995) used this metric (although defined over the Pareto optimal set, i.e. $|P_{known}|$).

It is difficult to determine what good values for $I_{ONVG}$ might be, because it may change at various computational resolutions. Reporting the ratio of $PF_{known}$'s cardinality to the discretized $PF_{true}$'s gives some feeling for the number of non-dominated vectors found versus how many exist:

$$I_{ONVGR} = | PF_{known} |/ |PF_{true}| \tag{48}$$

This metric, in addition to $I_{ME}$ and $I_{GD,}$ were proposed by Van Veldhuizen (1999) to be used conjointly, to compare MOEA performance.

**Strength and weaknesses:** Although counting the number of non-dominated solutions gives some feeling for how effective the MOEA is in generating desired solutions, it does not reflect on how "far" from $PF_{true}$ these non-dominated solutions are. As example, two approximation sets, one containing a million non-dominated solutions and the other containing just 1, with this point dominates all of those in the former set. No conclusion can be drawn regarding any dominance relation between two approximations sets (Knowles and Corne, 2002; Zitzler et al., 2003).

### 5.3.9 hypervolume indicator, $I_H$

Zitzler and Thiele (1998b) proposed a metric which calculates the hyper volume of the multi-dimensional region enclosed by $PF_{known}$ and a 'reference point' $z_{ref}$. In the two-dimensional case, each solution in $PF_{known}$ covers an area, a rectangle, defined by the point (0,0) and ($f_1(\underline{x})$, $f_2(\underline{x})$). The metric is then defined by:

$$I_H = \{ \bigcup_i v_i \mid \underline{x}_i \in P_{known} \} \tag{49}$$

where $v_i$ is the hypervolume defined by the reference point and the solution $\underline{x}_i$. In a minimisation and $z_{ref}$ as origin, lower values represent better sets ($I_H(A) < I_H(B)$) and in maximisation, the opposite.

**Strength and weaknesses:** The authors note that this metric may be misleading if $PF_{known}$ is non-convex. Moreover, the reference point is not always easy to determine (it must lie in a feasible region) and Knowles and Corne (2002) showed that this choice affect the ordering of non-dominated sets. It has a large computational overhead, $O(n^{k+1})$, rendering it unusable for large number of objectives or large sets. Although, this metric is scaling independent, it multiplies 'apples' by 'oranges'. But it is important to note that this metric is $\underline{\succ}$ compatible and $\triangleright$ complete provided that the reference point is set so that all feasible non-dominated sets are evaluated as positive (Knowles and Corne, 2002).

### 5.3.10 Spacing, $I_S$

This metric measures the spread (distribution) of vectors through $PF_{known}$. The approach based on niching and crowding attempt to spread each generation population evenly along the $PF_{current}(t)$. Schott (1995) proposed this metric called spacing to measure the range (distance) variance of neighbouring vectors in $PF_{known}$ (i.e. the diversity of the approximation set). He defined it as:

$$I_S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2} \tag{50}$$

where

$$d_i = \min_j\left(\left| f_1^i(\underline{x}) - f_1^j(\underline{x}) \right| + \left| f_2^i(\underline{x}) - f_2^j(\underline{x}) \right| + ... + \left| f_Z^i(\underline{x}) - f_Z^j(\underline{x}) \right|\right) \tag{51}$$

$\bar{d}$ is the mean of all $d_i$ and $n = |PF_{known}|$, $i,j = 1,2,...,n$. A value of zero for this metric indicates equidistant spaced members in $PF_{known}$.

**Strength and weaknesses:** If an MOP has a $PF_{true}$ composed of two or more Pareto curves, the distance between the end-points of two successive curves may skew this metric. So for this kind of Pareto sets, the distance corresponding to the "breaks" should be removed from the spacing computation. This implies the knowledge of the $PF_{true}$'s characteristics (Coello Coello et al., 2002). Moreover, Schott's definition does not specify the use of normalised distances, which may be problematic. This metric is neither compatible nor complete with respect to any dominance relation (Knowles and Corne, 2002; Zitzler et al., 2003) which make its use unreliable.

### 5.3.11 Chi-square-like deviation, $I_{CD}$

Srinivas and Deb (1994) defined a similar measure expressing how well an MOEA has distributed Pareto optimal solutions over a non dominated region. They defined this metric as:

$$I_{CD} = \left(\sum_{i=1}^{q+1}\left(\frac{n_i - \bar{n}_i}{\sigma_i}\right)^p\right)^{1/p} \tag{52}$$

where q is the number of desired optimal points and the (q+1)-th region is the dominated region, $n_i$ the actual number of individuals in the i-th sub region (niche) of the non dominated region, $\bar{n}_i$ is the expected number of individuals in the i-th sub region of the non dominated region, p=2, and $\sigma_i$ is the variance of individuals in the i-th sub region of the non dominated region. Using probability theory it was estimated elsewhere (Deb 1989) that

$$\alpha_i^2 = \bar{n}_i\left(1 - \frac{\bar{n}_i}{N}\right) \quad i = 1,2,..,q \tag{53}$$

where N is the population size. Since it is not desirable to have any individual in the dominated region ((q+1)-th sub region), $\bar{n}_{q+1}$ =0. If the distribution of points is ideal with $\bar{n}_i$ number of points in the i-th sub region, the performance measure will equal zero. Therefore, an algorithm with a good distributing capability is characterized by a low value of $I_{CD}$. This metric may be modified to measure the distribution of the solutions in objective space. As the $I_S$ indicator, it measures only the uniformity and complements the $I_{GD}$ or $I_{ME}$

**Strength and weaknesses:** Essentially, the same as for the $I_S$ indicator.

### 5.3.12 Maximum spread, $I_{MS}$:

This metric, called also the front spread, is proposed by Zitzler (1999) to indicate the size of the objective space covered by an approximation set. $I_{MS}(A)$ is defined to be the maximum Euclidean distance inside the smallest m-dimensional bounding box that contains A. This is computed using the maximum distance among the objective vectors members of A, in each dimension separately:

$$I_{MS}(A) = \sqrt{\sum_{i=1}^{m} \max_{\underline{x}^k, \underline{x}^l \in A} \left\{ (f_i(\underline{x}^k) - f_i(\underline{x}^l))^2 \right\}} \tag{54}$$

This metric can also be expressed in the decision variable space.

**Strength and weaknesses:** This diversity indicator, as the precedent is neither compatible nor complete with any relation of dominance. Moreover, it has the drawback of scaling dependent metrics.

### 5.3.13 Minimum distance between two solutions, $I_{MD}$

Sayin (2000), besides the cardinality and the coverage dimensions proposed a third one to evaluate the quality of an approximation set: the uniformity. The definition of the uniformity, given by the author, is: Let be Z a set $\subseteq IR^n$, and let be A a $d_\varepsilon$-representation of Z. A is called a $\delta$-uniform $d_\varepsilon$-representation of Z if $\min_{\underline{x},\underline{y} \in A, \underline{x} \neq \underline{y}} d(\underline{x},\underline{y}) \geq \delta$. The uniformity level is determined by the quantity:

$$I_{MD}(A) = \min_{\underline{x},\underline{y} \in A, \underline{x} \neq \underline{y}} d(\underline{x},\underline{y}) \tag{55}$$

A high value of $I_{MD}$ besides a high cardinality of A mean a high diversity of this approximation set.

**Strength and weaknesses:** Zitzler et al. (2003) advanced that when using this metric, as for other diversity indicators, no conclusion can be drawn regarding the dominance relationship between approximation sets.

### 5.3.14 Pareto spread, $I_{OS}$

Wu and Azarm (2001), proposed a metric to quantity the diversity of an approximation set. Two metrics are proposed: the overall Pareto spread, $I_{OS}$, which is identical to the hypervolume indicator, $I_H$, proposed by Zitzler and Thiele (1998b), and the $k^{th}$ objective Pareto spread indicator which provides an insight spread of the solutions' range with respect to each member of the objective vector. This latter is defined as:

$$I_{OS_k}(A) = \frac{\left|\max_{i=1}^{n}\{f_k(\underline{x}_i)\} - \min_{i=1}^{n}\{f_k(\underline{x}_i)\}\right|}{\left|f_k^* - f_{*k}\right|} \tag{56}$$

where $f_k^*$, $f_{*k}$ represent the ideal and anti-ideal points respectively.

### 5.3.15 Accuracy of the Pareto Frontier, $I_{AC}$

Three regions are defined relatively to a Pareto optimal set as shown in Figure 14: the inferior region, the non-inferior region and the dominant region.

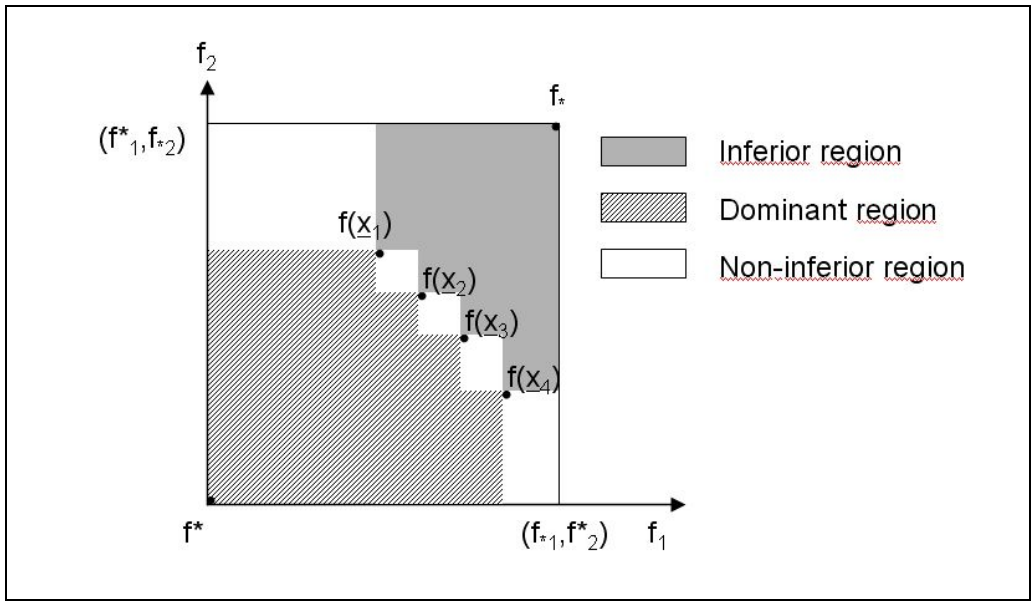


*Figure 14. Dominant, inferior, non-inferior region and observed Pareto frontier (PFknown) of an approximation set A in a case of minimisation*

Wu and Azarm (2001) defined an accuracy indicator of the observed Pareto frontier $PF_{known}$ by quantifying the region wherein it falls into. The impreciseness of the approximation set A comes from the fact that not all the points in the non inferior region can be simultaneously on the

approximation Pareto front. So the idea proposed by the authors is to quantify the normalised Pareto frontier of the approximation set A by:

$$AP(A)= 1- space(S_{inf}(A)) – space(S_{dom}(A)) \tag{57}$$

where $space(S_{inf}(A))$ and $space(S_{dom}(A))$ are the normalised hyper-volume of the Pareto set A's inferior and dominant regions, respectively. They are computed using the following relations:

$$space(S_{inf}(A)) = \sum_{r=1}^{|A|}\left\{(-1)^{r+1}\left[\sum_{k_1=1}^{|A|-r+1}\ldots\sum_{k_l=k_{l-1}+1}^{|A|-(r-l+1)+1}\ldots\times\sum_{k_r=k_r-1}^{|A|}\prod_{i=1}^{Z}\left[1-\max_{j=1}^{r}(\bar{f}_i(\underline{x}_{k_j})\right]\right]\right\} \tag{58}$$

$$space(S_{inf}(A)) = \sum_{r=1}^{|A|}\left\{(-1)^{r+1}\left[\sum_{k_1=1}^{|A|-r+1}\ldots\sum_{k_l=k_{l-1}+1}^{|A|-(r-l+1)+1}\ldots\times\sum_{k_r=k_r-1}^{|A|}\prod_{i=1}^{Z}\left[1-\min_{j=1}^{r}(\bar{f}_i(\underline{x}_{k_j})\right]\right]\right\} \tag{59}$$

with $\bar{f}_i$ the scaled $i^{th}$ component of the objective vector. The authors set the following normalisation (for a minimisation):

$$\bar{f}_i(\underline{x}) = \frac{f_i(\underline{x}) - f_j^*}{f_{j^*} - f_j^*} \tag{60}$$

The value of the quality metric "accuracy of the observed Pareto frontier" is then defined as:

$$I_{AC}(A) = 1 / AP(A) \tag{61}$$

When an approximation solution set is empty, the DM has the least (zero) amount of knowledge about the corresponding Pareto frontier. In this case according to Equation (61), AP(A) = 1 and AC(A) = 1. In the other extreme case, when the approximation set A contains all of the Pareto solutions and the Pareto frontier is continuous, then AP(A) = 0 and AC(A) = $\infty$. When comparing two observed approximation sets, the set with higher value of $I_{AC}$ is the preferred one.

**Strength and weaknesses:** The same as for the $I_{MD}$ indicator.

### 5.3.16 Number of distinct choices, $I_{NDC}$

This indicator of the diversity proposed by Wu and Azarm (2001) indicates how the solutions in the approximation set are distributed and measure their distinctiveness. The scaled objective space is divided into $1/\mu^Z$ small grids, $\mu$ ($0<\mu<1$) being a parameter defined by the DM. Each of the grids refers to a hypercube $T_\mu$ (in Z-dimension of the objective space) of size $\mu$, wherein any two solutions are considered similar by the DM (or that the DM is indifferent to such solutions).

The quantity $NT_\mu(q, A)$ indicates whether or not a solution of the approximation set falls in the hypercube Tμ, characterized by the point q. This quantity equals one if at least one solution exists otherwise it equals zero:

$$NT_\mu(q, A) = 1 \quad \exists\, \underline{x}_k \in A: f(\underline{x}_k) \in T_\mu(q) \tag{62}$$

$$NT_\mu(q, A) = 0 \quad \forall\, \underline{x}_k \in A: f(\underline{x}_k) \notin T_\mu(q) \tag{63}$$

The quality metric $I_{NDC,\mu}(A)$ is then :

$$I_{NDC,\mu}(A) = \sum_{l_m=0}^{\nu-1} \cdots \sum_{l_2=0}^{\nu-1} \sum_{l_1=0}^{\nu-1} NT_\mu(q, A) \tag{64}$$

where $q=(q_1,q_2,...,q_Z)$ with $q_i = l_i/\nu$ , $\nu=1/\mu$. The point q located at any intersection of Z-gridlines in the objective space, has coordinates $q=(q_1,q_2,...,q_Z)$. For pre-specified value of μ, an approximation Pareto set with higher value of $I_{NDC,\mu}$ is preferred.

**Strength and weaknesses:** This indicator is similar the Chi-square-like deviation metric proposed by Srinivas and Deb(1994).

### 5.3.17 Cluster, $I_{CL,\mu}$

Wu and Azarm (2001) pointed out that the quality metric defined above, $I_{NDC,\mu}$, cannot be used alone because the cluster phenomenon is not properly interpreted. For instance, for a pre-specified value of μ, two approximations set are generated, A which provides 10 distinct solutions with $I_{NDC,\mu}$ = 10 and B with 100 solutions and $I_{NDC,\mu}$ = 10. The second set, B, is not desirable by the DM since many of the solutions are likely to be clustered. Hence, the authors introduced:

$$I_{CL,\mu}(A) = \frac{|A|}{I_{NDC,\mu}(A)} \tag{65}$$

where |A| is the number of solutions observed in A. In the ideal case where every solution in the approximation set is distinct $I_{CL,\mu}(A)$ =1. In all other cases, $I_{CL,\mu}(A)$ >1. The higher the value of this metric is, the most clustered solution set is hence the less preferred this set is.

**Strength and weaknesses:** Essentially the same as others indicators of the diversity.

## 5.4 Performance assessment based on binary quality indicators

In contrast to unary indicators, only few binary indicators can be found in the literature. They can be used to overcome the difficulties described above with the unary indicators but they present themselves drawbacks. For example, when comparing *S* algorithms using binary indicators, we obtain $S(S\text{-}1)$ values- in contrast to the *S* values in the case of unary analysis. This renders the analysis and the presentation of the results more complicated. Moreover, Zitzler et al. (2003) demonstrated that a symmetric binary indicator (i.e., I(A,B) = c-I(B,A), c being a constant (Knowles and Corne, 2002)) can detect whether A is better than B, but not whether $A \succeq B$, A || B, or A=B. On the other hand, if $I(A,B) \neq 0$ for a pair A || B, $C_{I,E}$ cannot be $\triangleright$ -compatible if it is $\triangleright$ -complete.

### 5.4.1 Binary ε-indicator, $I_\varepsilon$

Zitzler et al. (2003) proposed a binary indicator which compares two sets using the dominance concept. They defined this metric as follows:

For any two approximation sets A and B and for ε>0,

$$I_\varepsilon(A,B) = \inf_{\varepsilon \in IR} \{ \forall \underline{x}^2 \in B \; \exists x^1 \in A : \underline{x}^1 \succeq_\varepsilon \underline{x}^2 \} \tag{66}$$

where $\underline{x}^1 \succeq_\varepsilon \underline{x}^2$ means that the objective vector $\underline{x}^1 = (x_1^1, x_2^1, ..., x_Z^1)$ is said to ε-dominates the objective vector $\underline{x}^2$ if and only if $\forall \; 1 \leq i \leq Z: \; x_i^1 \leq \varepsilon . x_i^2$ .

Accordingly, the ε-indicator gives the factor by which an approximation set is worse than another with respect to all objectives. This metric can be calculate in time O(Z. |A|. |B|) as follows:

$$\varepsilon_{\underline{x}^1, \underline{x}^2} = \max_{1 \leq i \leq Z} \frac{x_i^1}{x_i^2} \qquad \forall \underline{x}^1 \in A \, , \, \underline{x}^2 \in B \tag{67}$$

$$\varepsilon_{\underline{x}^2} = \min_{\underline{x}^1 \in A} \varepsilon_{\underline{x}^1, \underline{x}^2} \qquad \forall \underline{x}^2 \in B \tag{68}$$

$$I_\varepsilon(A,B) = \max_{\underline{x}^2 \in B} \varepsilon_{\underline{x}^2} \tag{69}$$

or equivalently

$$I_\varepsilon(A,B) = \max_{\underline{x}^2 \in B} \min_{\underline{x}^1 \in A} \max_{1 \le i \le Z} \frac{x_i^1}{x_i^2} \tag{70}$$

the interpretation function E:=( $I_\varepsilon(A,B) \le 1 \wedge I_\varepsilon(B,A) > 1$) $\Leftrightarrow A \rhd B$. This method of comparison is both $\rhd$ -complete and $\rhd$ -compatible. Moreover, the authors showed that this metric is also complete and compatible for the other performance relations:

$$E:=(I_\varepsilon(A,B) \le 1) \Leftrightarrow A \succeq B, \tag{71}$$

$$E:=(I_\varepsilon(A,B) = 1 \wedge I_\varepsilon(B,A) = 1) \Leftrightarrow A = B, \tag{72}$$

$$E:=(I_\varepsilon(A,B) > 1 \wedge I_\varepsilon(B,A) > 1) \Leftrightarrow A \| B. \tag{73}$$

An equivalent additive ε-indicator has been defined in the same manner.

**Strength and weaknesses:** This metric is relatively easy to calculate and is complete and compatible with the relations $\rhd$ , $\succeq$ , =, and || . However, its use is restricted to some conditions: it is limited to the use of comparison of two approximation sets rather than two algorithms, i.e. for each EA algorithm, only one run is performed.

### 5.4.2 Two set coverage, $I_{CS}$

This metric indicates if an approximation set A "covers" another set B by calculating the number of solutions in B that are dominated by at least one solution $\in$ A. This quality indicator is complete and compatible for the following performance relations:

$$E:=(I_C(A,B) = 1 \wedge I_C(B,A) = 0) \Leftrightarrow A \succ B \tag{74}$$

$$E:=(I_C(A,B) = 1 \wedge I_C(B,A) < 1) \Leftrightarrow A \rhd B \tag{75}$$

$$E:=(I_C(A,B) = 1) \Leftrightarrow A \succeq B \tag{76}$$

$$E:=(I_C(A,B) = 1 \wedge I_C(B,A) = 1) \Leftrightarrow A = B \tag{77}$$

$$E:=(0 < I_C(A,B) < 1 \wedge 0 < I_C(A,B) < 1) \Leftrightarrow A \| B \tag{78}$$

### 5.4.3 Utility indicators, $I_{R1}$, $I_{R2}$, $I_{R3}$

Hansen and Jaszkiewcz (1998) proposed three symmetric binary indicators that are based on a set of utility functions.

$$I_{R1}(A,B,U,p)= \int_{u\in U} C(A,B,u)p(u)du \tag{79}$$

$$\text{with } C(A,B,u)=\begin{cases}1 & if\ u^*(A)>u^*(B)\\ \frac{1}{2} & if\ u^*(A)=u^*(B)\\ 0 & if\ u^*(A)<u^*(B)\end{cases} \tag{80}$$

where U is some set of utility functions, u : $IR^Z \rightarrow IR$ which maps each points in the objective space into a measure of utility, p(u) is an intensity function expressing the probability density of the utility $u \in U$, and $u^*(A) = \max_{z\in A}\{u(z)\}$ and similarly for u*(B). $I_{R1}$ is based on calculating the probability that A is better than B over a set of utility functions. It's a direct comparative metric and, when used with a reference set, induces a total ordering (Knowles and Corne, 2002). The authors introduced another metric that measures how much A is better than B.

$$I_{R2} = \int_{u\in U} (u^*(A) - u^*(B))p(u)du \tag{81}$$

The third indicator, $I_{R3}$, is similar to $I_{R2}$ but the ratio of the best utility values is calculated instead of the difference.

**Strength and weaknesses:** Knowles and Corne (2002) showed that IR1 is cycle inducing and these three indicators depend upon the ability to define a set of utility functions incorporating preference information. The compatibility with $\succeq$ relation is subject to a set of conditions including the condition that the utility functions are strictly compatible with the dominance relation.

Zitzler et al.(2003) indicated that these metrics are $\rhd$ complete but not $\rhd$ compatible because they are symmetric and I(A,B) can be greater or less than 0 if A || B (in reference to their theorem 3).

### 5.4.4 Lines of intersection, $I_{L1}$

Knowles and Corne (2000b) proposed a comparison method based on the study of Fonseca and Flemming (1996) designed for the statistical analysis of multiple runs of an algorithm. The method uses a user-defined set of lines in the objective space, passing from the origin and none of them is perpendicular to any of the axes. For each line, the intersections with the attainment surface (Fonseca and Flemming, 1996) represented by the approximation set $PF_{known}$ are calculated. Theses intersections are then sorted according to their distance from the origin, and the resulting order defines a ranking of the approximation sets with respect to this line. If only

two approximation sets are considered, the metric $I_{L1}(A,B)$ gives the fraction of the lines for which A is ranked higher than B and $I_{L1}(B,A)$ the fraction of the lines for which B is ranked higher than A. A more common result in practice is that the two values sum to rather less than 100% indicating that no significant conclusion can be made with respect to many of the sampling lines. The most significant outcome would be $I_{L1}(A,B) = 1$ and $I_{L1}(B,A) = 0$.

**Strength and weaknesses:** This method strongly depends on the number of lines chosen and certain parts of $PF_{known}$ can not be sampled. As a consequence, this comparison method $C_{I_{L1},E}$ with $E := (I_{L1}(A,B) = 1 \wedge I_{L1}(B,A) = 0)$ is not $\rhd$ compatible; it is, however, $\not\rhd$-compatible and $\succ\!\!\succ$-complete (Zitzler et al., 2003).

## 5.5 What if out performance does not hold?

In this work, $I_\varepsilon$ is calculated after objectives to be maximised are transformed. If no conclusion could be drawn from the comparison (incomparable strategies) we can examine the feature of the algorithms regarding the convergence (proximity) and the diversification by using the following indicators, assuming that no a priori information is available on the Pareto set and taking into account the stochastic aspect of the methods. For the proximity, we have chosen the similarity to ideal solution index, **C***, proposed by Hwang and Young (1981) in their multiple attribute decision making procedure called Technique of Order Preference by Similarity to Ideal Solution (TOPSIS).

*Similarity to the ideal solution*: The closeness of the approximation set **S** to the Pareto front set could be represented by the similarity of the centroid of S to the ideal solution:

$$C^* = \frac{1}{N_s}\sum_i C_i^* \tag{82}$$

$$\text{where } C_i^* = S_i^- /(S_i^* + S_i^-) \text{ and } S_i^* = \sqrt{\sum_{j=1}^{Z}(\nu_{ij} - \nu_j^*)^2} \text{ and } S_i^- = \sqrt{\sum_{j=1}^{Z}(\nu_{ij} - \nu_j^-)^2} \tag{83}$$

where $\nu_j^*$ and $\nu_j^-$ are respectively the best and the worst values of the normalized objectives $\nu_{ij}$, with $1 \le j \le Z$ and $1 \le i \le T_s$, $T_s$ being the total number of efficient solutions belonging to the set representing the union of all the approximation sets to be compared simultaneously, and $\nu_{ij} = \dfrac{f_j(x_i)}{\sqrt{\sum_i f_j^2(x_i)}}$. Note that, unlike in the original work, no weight is associated to the objectives $f_j$. The ideal solution is represented by $A^* = \{\nu_1^*, \nu_2^*, .., \nu_j^*, .., \nu_Z^*\}$ and the anti-ideal solution $A^- = \{\nu_1^-, \nu_2^-, .., \nu_j^-, .., \nu_Z^-\}$. $0 \le C^* \le 1$ where $C^* = 0$ for $A^-$ and $C^* = 1$ for $A^*$.

C* is similar to the proximity indicators proposed in the literature such as the generational distance proposed by Van Veldhuizen (1999) which expresses the average distance to the Pareto

front or the distance from a reference set proposed by Czyzak and Jaszkiewicz (1998). For the diversification, we propose three indicators that, in our opinion, reflect well this property: the cardinality of the approximation set $\mathbf{N_s}$, the diversity of solutions in the decision space $\mathbf{D_s}$, the spread of solutions on the Pareto front **COV**.

*Cardinality of the Pareto set*: The non-dominated solutions are stored in an external list that is updated at each iteration. Thus, the approximation Pareto set **S** is represented by this population when the stopping criterion is reached.

*Diversity of the Pareto set*: When designing a GA algorithm, the diversification of the population (solutions) is often the major goal through the exploration and exploitation operators. Ultimately, the diversity of the solution in the Pareto set represents a desirable result as it gives to the decision maker a wider range of potential choices. The diversity of a population used here is inspired from the Shannon theory to evaluate the value of information based on the entropy H. A higher the entropy of a message ($0 \leq H \leq 1$) implies a lower the redundancy of the information and a higher the value of this message.

The diversity of the solutions is based on the number of alternatives of resource combinations that could be allocated for each task i in the population. So the total number of COA variants is given by the number of different resource combinations for all the tasks.

$$D_s = \frac{1}{n}\sum_{i=1}^{n} H_i \tag{84}$$

$$\text{with } Hi = \frac{1}{\log(q_i)}\sum_{j=1}^{q_i}\frac{nij}{popsize}\log\left(\frac{nij}{popsize}\right) \tag{85}$$

where $D_s$ is the diversity of the population, $H_i$ the entropy of a task i, n the number of tasks in the COA, $q_i$ the number of resource combinations per task found in the population, and $n_{ij}$ the number of solutions (COAs) with different combinations of resources for task i. The higher the number of COAs with different resource configurations per task, the greater the diversity of the solutions set.

*Diversity of the Pareto front*: We propose expressing the diversification of the solutions spread all over the Pareto frontier by the extent of coverage in each objective's dimension separately as illustrated in Figure 15. This indicates the size of the objective space covered by the Pareto set. The extent of coverage relative to each objective $f_i$ is defined as follows:

$$Cov_i = \frac{\max\limits_{x,y\in S}\left|f_i(x) - f_i(y)\right|}{\left|f_i^* - f_{i*}\right|} \text{ x100} \tag{86}$$

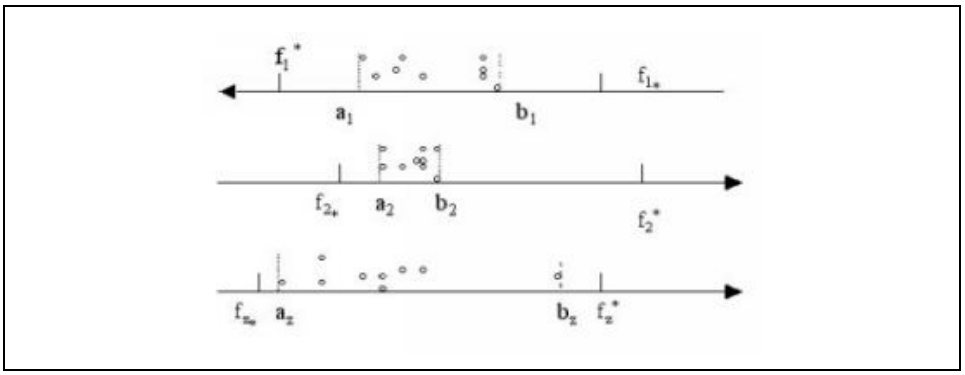


*Figure 15. Spread of solutions over the Pareto front in multidimensional space*

where the numerator expresses the maximum distance among solutions of the approximation set **S** in the dimension i, and $f_{i*}$ and $f_i^*$ are, respectively, the anti-ideal and ideal values of $f_i$ (the extremes of the objective or its worst and best values among efficient solutions belonging to the union of all the approximation sets to be compared).

Zitzler and Thiele (1998), Bosman and Thierens (2003) have proposed similar indicator in which the $Cov_i$ for $i$= 1,…, $Z$, are aggregated by considering the union of all bounding-boxes covered by the Pareto set. We consider that this is not very convenient because a single crisp global score for the coverage does not reflect how is the diversification with respect to a given criteria. For a better appreciation of the dispersion, it is more appropriate to consider the coverage array on all the objectives:

$$COV = (Cov_1, Cov_2, \ldots Cov_Z) \tag{87}$$

Evaluating an algorithm's performance can be achieved by comparing its outcome (an approximation set) to the true Pareto optimal set, if this is available, or to another algorithm's outcome. The most general statements possible are of the form "approximation set A strictly dominates/dominates/is better than/ weakly dominates approximation set B" which are based on the concept of Pareto dominance. A rigorous comparison method should be at least capable of detecting whether A is better than B, hence should be compatible and complete with respect to as many dominance relations as possible (Ziztler et al., 2003). These authors found that most of unary quality metrics used to evaluate these algorithms are not capable of indicating whether an approximation set is better than another, even if several of them are used simultaneously. Some of them allow almost inferring that an approximation set is not worse than another, such as the distance indicator (Czyzak and Jaszkiewicz, 1998), and the hypervolume indicator (Zitzler and Thiele,1998b). The few binary quality indicators proposed in the literature, which assign real numbers to ordered pairs of approximation sets, are capable of detecting whether an approximation set is better than another (e.g, binary ε-indicator by Zitzler et al. (2003)). However their great inferential power comes with some limitations such as the comparison algorithm's outcome from single run (do not take into account the stochastic aspect of these algorithms) and the complexity of the comparison method: the number of indicator values to be considered is not linear (as the unary indicators) but quadratic in the number of approximation sets.

As pointed out by Bosman and Thierens (2003), the comparison between outcomes of MOEAs do not always lead to a clear outperformance of one algorithm over an other, as the efficiency of newly designed MOEAs increases. Incomparable approximation sets often occur but this does not imply that an algorithm is not preferred to another. Even if A || B, two approximation sets generated by algorithm A and B respectively, we could still prefer A to B for their diversity or convergence features. So the quality metrics reported above can offer additional information to decide which algorithm puts emphasis on diversity preservation or getting as close as possible to the Pareto optimal front.

# 6 Planning courses of action: computational results

In this section the performance of the proposed genetic algorithms with the different replacement procedures is examined. The GAs are implemented in C++, using a DLL (Dynamic link library) to integrate the first order Multicriteria Filtering Procedure (PFM) which has been coded in Visual Basic. Moreover, the performance of our metaheuristic is compared to other efficient multiobjective evolutionary algorithms (MOEAs) of the literature. To take into account the stochastic aspect of these MOEAs, several runs are performed. The comparison is done using performance indicators presented in section 5. When it is necessary, the significance of theses indicators is examined by using the Wilcoxon signed-rank statistical test since the data distributions are not necessarily symmetrical.

In order to evaluate the relative performance of the methods, we first check if a Pareto approximation set obtained from one method weakly dominates a set obtained by the other one. This set domination is defined as:

$S_1$ weakly dominates $S_2$ if $\forall x \in S_2, \forall y \in S_1, \forall i, 1 \le i \le Z, f_i(y) \ge f_i(x)$ (88)

Then we compare the difference between their performance indicators and we test their significance by using the Wilcoxon signed-rank statistical test since the data distributions are not necessarily symmetrical. If no conclusion could be drawn, we can examine which method puts more emphasis on diversity or on proximity.

The efficiency of the method proposed here is investigated using examples of courses of action with four objectives: the cost and the make-span to be minimised (linear objectives), the resource reliability and the impact on the enemy, to be maximised. The impact is measured on a qualitative scale (ordinal objective). Resource availability and task precedence constraints are considered. Three examples are studied to examine the effect of the problem size. The performance indicators of an algorithm are compiled from 5 to 10 runs of each test application. Comparison between algorithms is done based on the means of these indicators. Since the objectives are the cost, the make-span, the reliability, and the impact, the diversity COV of the approximated Pareto set is defined as: COV = ($cov_{cost}$, $cov_{reliabilty}$, $cov_{impact}$, $cov_{makespan}$).

In the first example, we consider 6 tasks and 5 generic (types) resources, combined in different sets, to generate randomly 10 initial COAs. This example is denoted by **6t-problem**. The resources are defined by their cost (fixed and in-use cost) and their reliability used to compute the cost and reliability of the COA (Guitouni et al,. 2002). The impact of a COA is calculated as the median value of the actions' impact, which are ordinal values given by the decision makers (Guitouni et al., 2002). In this study, these data are randomly generated. The make-span considered here is the total delay between all the actions in the COA network. In the second example, we consider courses of action defined by 50 tasks and 3 generic resources and denoted by **50t-problem.** In the third example, denoted by **100t-problem,** 100 tasks and 3 generic resources are considered. Initial populations of 15 and 22 COAs are created for the **50t-problem** and **100t-problem** respectively, by using a heuristic based on the network approach and CPlex (Urli et al., 2003). The common parameters used in MFGA are:

– probability of crossover $p_c = 0.6$,

– probability of mutation $p_m = 1/popsize$ which gives a better result compared to a fixed value.

– population size returned by the filtration procedure $P_f$.

– stopping criteria $gen_{max}$ = 500 generations (6t-problem), 250 generations (50t-problem), 150 generations (100t-problem).

## 6.1 Impact of population size

In this section, we have examined the influence of the population size parameters Pc and Pf on the algorithm's performance (see Figure 16). In Figure 16, we compare the results of the GA process when PFM1 (strategy 1) is used as replacement procedure for the 6t-problem and for two pair of (Pc,Pf ) values. It is shown clearly that a couple-value of (90, 30) gives a better evolution of the population fitness than (150, 50). One can see also in Figure 15 (h) that the search began in the feasible region, crossed the unfeasible region to attain more interesting neighbourhoods where others new feasible solutions were discovered. This powerful research strategy is attributed to the formulation of the vector fitness that integrates both the objectives and the constraints. In this way, the search is enforced towards near-optimal solutions from both sides of feasible and infeasible regions.

Several values of Pc and Pf are tested. We compare the approximation sets (five runs for each test application) obtained when the NDS replacement procedure is used for the 6t-problem. This comparison is based on the binary $\varepsilon$-indicator, $I\varepsilon$, calculated for all the tests (Table 4). The results show that the test application for which the population size is increased to Pc = 30 individuals before being filtered to 10 individuals is the worst case. $\forall$ B corresponding to the case (40 $\rightarrow$20) or (60 $\rightarrow$20) or (60 $\rightarrow$30) or (90$\rightarrow$-30) or an evolution strategy with a constant population size P*=30, we have $I\varepsilon$(B, 30$\rightarrow$10) = 1.0 $\wedge$ $I\varepsilon$(30 $\rightarrow$10,B) > 1 (see section 5.5). However, we can not conclude about the remaining case studies as $I\varepsilon$(B, A) > 1 $\wedge$ $I\varepsilon$( A,B) > 1.

If we examine performance regarding the proximity and the diversity, as shown in Table 5, that a filtered population size of 30 gives the best results in terms of diversity with higher Ns, Ds and COV and in term of proximity. This $P_f$ value corresponds by this way to the size of the problem (6 tasks$\times$5 combinations of resources). Before filtering, doubling (60$\rightarrow$ 30) or tripling (90$\rightarrow$ 30) the population appear to be equivalent although a better coverage is achieved in the latter case. Similar conclusions are derived when the two other replacement procedures (MFP and NDS+MFP) were simulated.

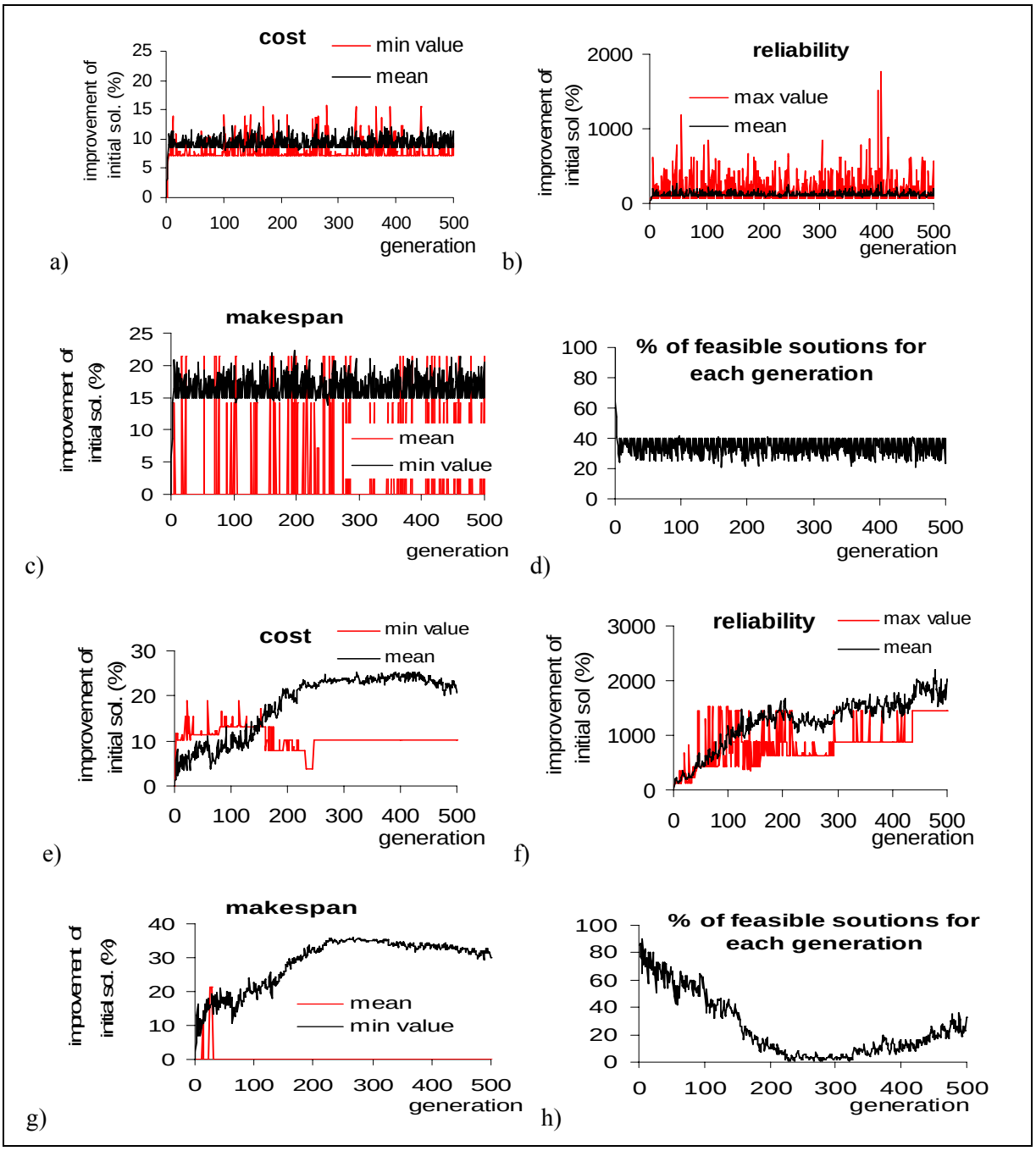


*Figure 16. Qualitative progress of the optimization process for the 6t-problem using PFM1 as replacement procedure. a),b),c),d): Pc =150 and Pf =50; e),f),g),h): Pc =90 and Pf =30*

*Table 4. The binary ε-indicator value $I_\varepsilon(A,B)$ for different critical population size $P_c$ (Test application: strategy 2 for the replacement procedure in the 6t-problem)*

| **$P_c$->$P_f$** | **A** | | | | | |
|---|---|---|---|---|---|---|
| **B** | **30 → 10** | **42 → 20** | **60 → 20** | **60 → 30** | **90 → 30** | **P*=30** |
| 30 → 10 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 40 → 20 | 2.0 | 1.0 | 1.1 | 1.2 | 1.07 | 1.08 |
| 60 → 20 | 2.0 | 1.3 | 1.0 | 1.1 | 1.2 | 1.0 |
| 60 → 30 | 2.0 | 1.2 | 1.6 | 1.0 | 1.2 | 1.2 |
| 90 → 30 | 2.0 | 1.5 | 1.9 | 1.1 | 1.0 | 1.1 |
| P*=30 | 2.0 | 1.5 | 1.9 | 1.1 | 1.2 | 1.0 |

P*: constant size at each generation

*Table 5. Pop. size effect on the proximity versus diversity for the tests where incomparable sets are obtained (Test application: strategy 2 for the replacement procedure in the 6t-problem)*

| | **Diversity** | | | **Prox** |
|---|---|---|---|---|
| **Pc→Pf** | **Ns** | **Ds** | **COV** | **C*** |
| 40→20 | 2.2 | 0.59 | (4.6; 6.6; 5; 10 ) | 0.45 |
| 60→20 | 5.4 | 0.50 | (10; 8.2; 25; 24 ) | 0.38 |
| 60→30 | 3.7 | 0.70 | (20; 14.6; 0; 18) | 0.46 |
| 90→30 | 4.4 | 0.69 | (23; 12; 20; 36) | 0.46 |
| P* = 30 | 3.5 | 0.73 | (13; 11.5; 30; 11.3) | 0.50 |

This study is a guideline about the population size to choose as input and output parameters of the GA replacement procedure. A population size equals to the problem size appears to be adequate for the parameter $P_c$ and increasing threefold this population size ($P_f$) before applying again the replacement procedure seems to be a good strategy.

## 6.2 Comparison between the replacement strategies

First, we compare performance of the three replacement strategies proposed for our mGA (mGA-MFP, mGA-NDS, mGA-NDS+MFP) using the 6t-problem as test application. The critical size of the population selected to apply the replacement strategy is Pc = 90 and the population is filtered to a size Pf = 30. This choice is based on the empirical results presented in Table 4. The approximation sets, obtained over 10 runs (after the dominated solutions have been discarded), are compared based on the binary ε-indictor (Table 6). The results show that:

$$I_\varepsilon(A_{MFP},B_{NDS}) > 1 \ \wedge \ I_\varepsilon(B_{NDS},A_{MFP}) = 1 \tag{89}$$

$$I_\varepsilon(A_{MFP},C_{MFP+NDS}) > 1 \ \wedge \ I_\varepsilon(C_{MFP+NDS},A_{MFP}) = 1 \tag{90}$$

$$I_\varepsilon(B_{NDS},C_{MFP+NDS}) > 1 \ \wedge \ I_\varepsilon(C_{MFP+NDS},B_{NDS}) = 1 \tag{91}$$

*Table 6. Comparison between the approximation sets obtained from the three replacement strategies based on the binary ε-indicator in the 6t-problem*

| $I_\varepsilon$ | $A_{MFP}$ | $B_{NDS}$ | $C_{MFP+NDS}$ |
|---|---|---|---|
| $A_{MFP}$ | 1.0 | 1.0 | 1.0 |
| $B_{NDS}$ | 2.0 | 1.0 | 1.0 |
| $C_{MFP+NDS}$ | 2.0 | 1.4 | 1.0 |

Thus, we can conclude that the replacement strategy 3 (MFP+NDS) clearly outperforms the MFP and the NDS procedures as $C_{MFP+NDS} \triangleright B_{NDS} \triangleright A_{MFP}$. If we look to the diversity and the proximity that characterize the sets obtained from the three strategies on Table 7, we can notice that the MFP+NDS replacement procedure gives a higher number of non-dominated solutions $N_s$, a higher diversity in decision space Ds, and greater coverage COV. The sets are also closer to the ideal solution. The multi-criteria-filtering procedure does not generate all non-dominated solutions because some of these solutions are eliminated by the conjunctive procedure. In the NDS replacement method, among the $P_f$ solutions filtered, those which have the same rank (thus incomparable between them) are selected randomly with respect to their position in the list. By contrast in the MFP+NDS replacement strategy, these solutions are selected in accordance with their objectives by the conjunctive and disjunctive methods even if globally they are incomparable (not dominated). Such a strategy leads to a population of better quality, as their evaluations are greater than the threshold values. . MCFP+NDS strategy is then retained as replacement procedure in MFGA.

*Table 7. mGAs performance (diversity and proximity) using the three replacement strategies in the 6t-problem*

| | Diversity | | | Prox |
|---|---|---|---|---|
| **strategy** | **Ns** | **Ds** | **COV** | **C*** |
| MFP | 3.0 | 0.56 | (12.2; 5; 12.5; 25) | 0.29 |
| NDS | 4.4 | 0.69 | (23; 12; 20; 36) | 0.40 |
| MFP+NDS | 6.0 | 0.84 | (34; 26.6; 20; 36.4) | 0.45 |

## 6.3 Comparison with efficient MOEAs of the literature

The performance of MFGA is compared to other efficient MOEAs from the literature. The first algorithm is the elitist non-dominated sorting genetic algorithm method (ENGA) proposed by Bagchi (1999) which has been found more efficient to discover the Pareto front compared to the well-known NSGA (Srinivas and Deb, 1994) for multi-objective scheduling problems. This algorithm uses domination rank and elitism to achieve better proximity and niche formation and sharing fitness for a better diversity of the Pareto front. The second algorithm is the fast elitist non-dominated sorting genetic algorithm NSGAII (Deb et al., 2000a and 2000b) which uses the domination rank to achieve better proximity and the binary tournament selection based on rank and crowding for a better diversity of the Pareto front. The third algorithm is the strength Pareto evolutionary algorithm SPEA2 (Zitzler et al., 2001) which uses an enhanced archive truncation method to achieve convergence and the binary tournament selection based on domination strength and nearest neighbourhood density estimation technique for a better diversity of the Pareto front. In order to compare the performance of the selection procedures of these algorithms, therefore their search procedure, the crossover and mutation operators are the same for all the methods.

### 6.3.1 Results for the 6t-problem

For MFGA, the critical size of the population selected to apply the replacement strategy is $P_c = 90$ and the population is filtered to a size $P_f = 30$. This choice is based on the empirical results presented in Table 5. For the ENGA algorithm, the population size is fixed to 30 (such as the Pf value in MFGA) and the niche size, $\sigma_{share}$, is set to a value of 5 which gives the best results after several trials between 1 and 5. As stated by the author, the sharing distance is computed in the phenotype space and the coefficient $\alpha$ of the sharing function $\alpha$=2. Ten runs are compiled as well for each test application. For NSGAII and SPEA2, the only required parameter is the population size that is set to 30.

MFGA works better than ENGA (Bagchi, 1999) as shown in Figure 17. ENGA is not able to maintain a high number of feasible non-dominated solutions, from one iteration to another one, for such a highly constrained problem. Only one run over five generated an interesting solutions set. The constraints in the fitness vector are not always fulfilled: they do not reach their maximum value (see section 6.2) even if the solutions are non-dominated having the best-scored dummy

fitness. In MFGA, feasible non-dominated solutions are archived and updated at each generation. This prevents the loss of interesting solutions during the algorithm processing. The comparison with NSGAII and SPEA2 (for ten runs also) using the $\varepsilon-$indicator shows that NSGAII outperforms SPEA2 and MFGA. The following relationship might be deduced from table 8:

$$\begin{aligned} &I\varepsilon(A_{NSGAII}, B_{SPEA2}) = 1.2 \wedge I\varepsilon(B_{SPEA2}, A_{NSGAII}) = 2 \Leftrightarrow A_{NSGAII} \parallel B_{SPEA2}; \\ &I\varepsilon(A_{NSGAII}, C_{MFGA}) = 1 \wedge I\varepsilon(C_{MFGA}, A_{NSGAII}) = 250 \Leftrightarrow A_{NSGAII} \rhd C_{MFGA}; I\varepsilon(B_{SPEA2}, \\ &C_{MFGA}) = 1 \wedge I\varepsilon(C_{MFGA}, B_{SPEA2}) = 250 \Leftrightarrow B_{SPEA2} \rhd C_{MFGA} \end{aligned} \quad (92)$$

This leads to the following conclusion for the 6t-problem:

$$(A_{NSGAII} \parallel B_{SPEA2}) \rhd C_{MFGA} \quad (93)$$

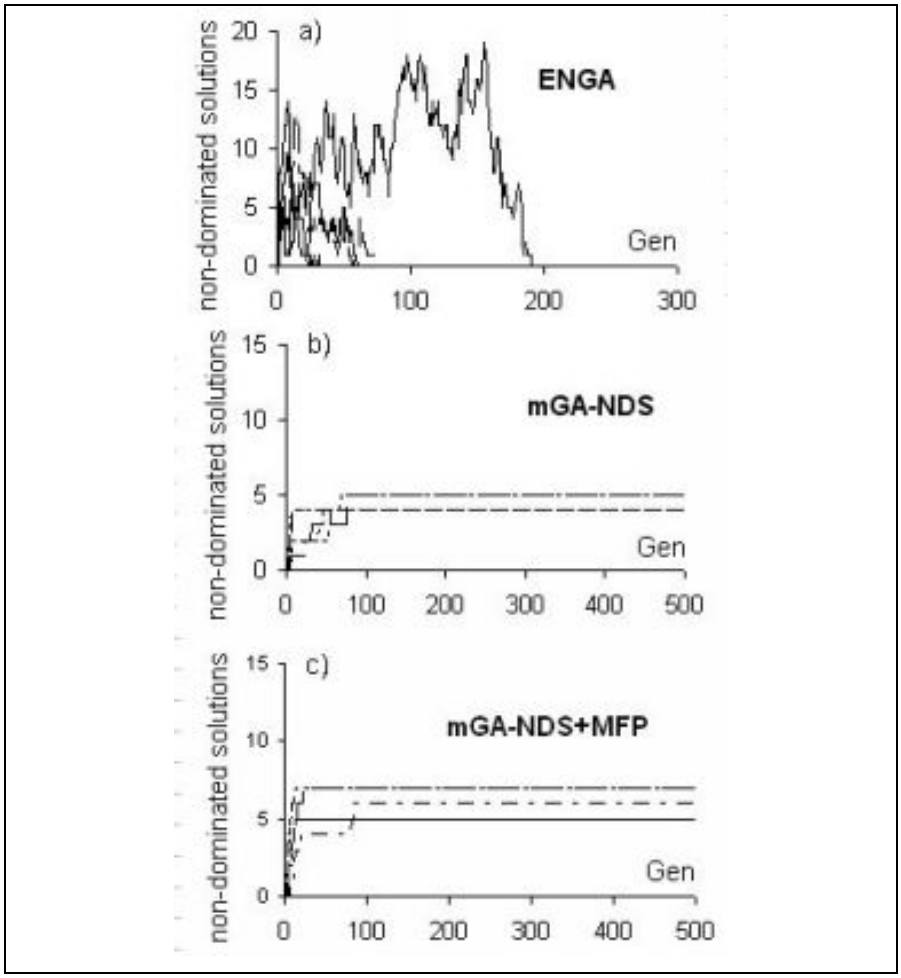


*Figure 17. Progress in finding non-dominated solutions for the 6t-problem. 5 runs are compiled for each method.*

*Table 8. Comparison between the three MOEAs for the 6t-problem*

| **CPU(sec)** | **22** | **41** | **31** |
|---|---|---|---|
| **Iε** | **$A_{NSGAII}$** | **$B_{SPEA2}$** | **$C_{MFGA}$** |
| $A_{NSGAII}$ | 1 | 2 | 250 |
| $B_{SPEA2}$ | 1.2 | 1 | 250 |
| $C_{MFGA}$ | 1 | 1 | 1 |

### 6.3.2 Results for the 50t-problem

The efficiency of these MOEAs is also examined for large problem sizes that are more constrained. For the 50t-problem, Pc = 150 and Pf = 50 as parameters' values give the best simulations. This confirms the empirical result found in Section 6.1 which states that choosing Pf equals to the size problem (50 tasks and 3 generic resources) and Pc= 3*Pf is appropriate. The ENGA algorithm, as for the 6t-problem, failed to maintain the non-dominated solutions up to the end of the GA processing. Nevertheless, to compare the sharing fitness approach used in ENGA method, we have modified the former by archiving and updating the approximation Pareto set discovered at each generation. For the three algorithms, ENGA, NSGAII, and SPEA2, the population size is set to 50 and the niche size $\sigma_{share} = 5$ for ENGA. The comparison is done over ten runs for each algorithm and $I\varepsilon$ is used as the performance indicator. Table 9 summarizes the results of the comparison. For this problem size we have:

$$(A_{NSGAII} \,||\, C_{MFGA}) \rhd (B_{SPEA2} \,||\, D_{m\text{-}ENGA}) \tag{94}$$

NSGAII and our algorithm generate incomparable approximation sets since we have

$$I\varepsilon(A_{NSGAII}, C_{MFGA}) > 1 \;\wedge\; I\varepsilon(C_{MFGA}, A_{NSGAII}) > 1. \tag{95}$$

However, both algorithms outperform SPEA2 and the modified ENGA. If we look at the quality of the outcomes of MFGA and NSGA, we can see in Figure 18 that the solutions generated by the former are more clustered near the ideal point defined here by [0, 1, 1, 0] for scaled objectives of cost, reliability, impact and makespan respectively; whereas the solutions generated by NSGAII are more scattered. This is confirmed by the proximity indicator C* value given in Table 10 which represents the average of the approximation sets proximity values calculated over ten runs. The coverage in the objective space is higher for the MFGA outcomes but the diversity in the decision space and the cardinality of the NSGAII outcomes are higher. The extent of the coverage on the third objective (the impact) is nil ($cov_{impact} = 0$) for this large-problem size. This result is the consequence that all non-dominated solutions always score the median value for this objective. We underline also that our algorithm is more CPU time consuming.

*Table 9. Comparison between the four MOEAs for the 50t-problem*

| **CPU(sec)** | **759** | **600** | **1331** | **1363** |
|---|---|---|---|---|
| **Iε** | **$A_{NSGAII}$** | **$B_{SPEA2}$** | **$C_{MFGA}$** | **$D_{m\text{-}ENGA}$** |
| $A_{NSGAII}$ | 1 | 10.5 | 1.1 | 6.3 |
| $B_{SPEA2}$ | 1 | 1 | 1 | 1.2 |
| $C_{MFGA}$ | 56693 | 600000 | 1 | 360000 |
| $D_{m\text{-}ENGA}$ | 1 | 1.7 | 0.99 | 1 |

*Table 10. Algorithms performance comparison regarding the diversity and the proximity of their incomparable approximation sets*

| **50t-problem** | **Diversity** | | | **Proximity** |
|---|---|---|---|---|
| **strategy** | **Ns** | **Ds** | **COV** | **C*** |
| NSGAII | 24.6 | 0.84 | (45; 73; 0; 61) | 0.44 |
| MFGA | 14.2 | 0.76 | (68; 94 ; 0; 34) | 0.84 |

### 6.3.3 Results for the 100t-problem

For the 100t-problem, $P_c = 150$ and $P_f = 50$ for MFGA yield the best compromise in terms of CPU time even if higher $P_f$ values generate higher $N_S$. For the sake of comparison, a population size equals to 50 is chosen for the three other MOEAs. The comparison of the outcomes of these algorithms using $I\varepsilon$ is given on Table 11. For this problem size, our algorithm outperforms all other MOEAs:

$$C_{MFGA} \succ\succ (D_{m\text{-}ENGA} \parallel A_{NSGAII}) \rhd B_{SPEA2} \quad (96)$$

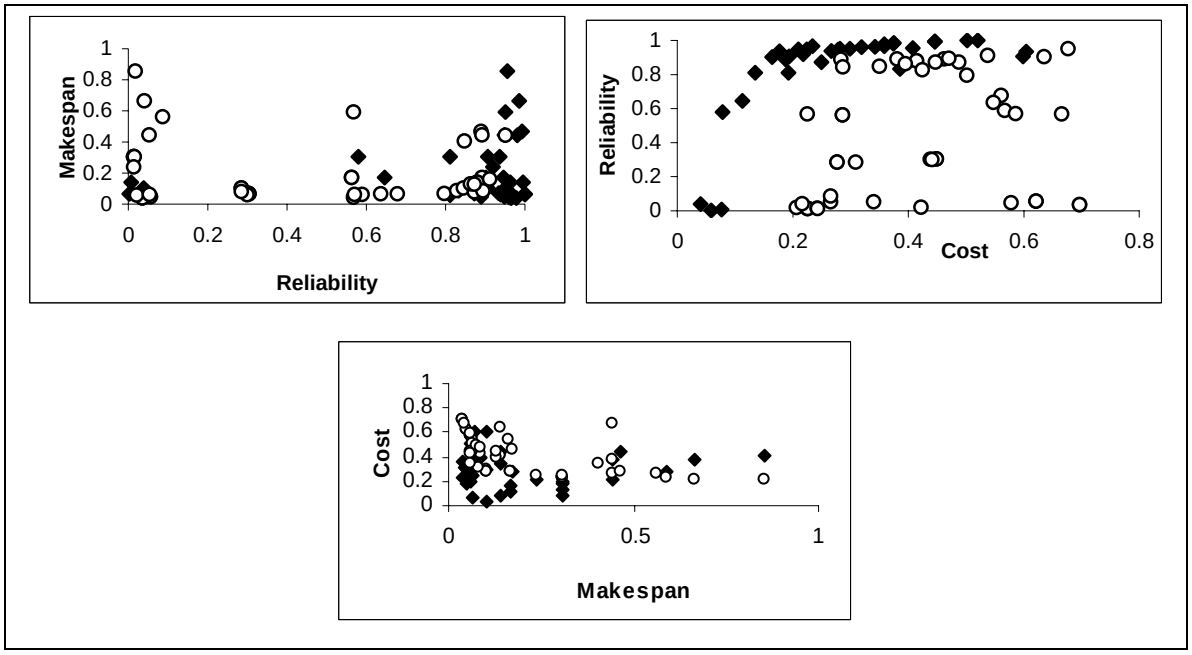


*Figure 18. Qualitative comparison between incomparable NSGAII outcome (empty points) and MFGA outcome (filled points) for the 50t-problem.*

*Table 11. Comparison between the four MOEAs for the 100t-problem*

| **CPU(sec)** | **2100** | **1149** | **2963** | **3420** |
|---|---|---|---|---|
| **Iε** | $\mathbf{A_{NSGAII}}$ | $\mathbf{B_{SPEA2}}$ | $\mathbf{C_{MFGA}}$ | $\mathbf{D_{m\text{-}ENGA}}$ |
| $A_{NSGAII}$ | 1 | 1.8 | 0.99 | 1.1 |
| $B_{SPEA2}$ | 1 | 1 | 0.99 | 1 |
| $C_{MFGA}$ | 3.2 | 5.8 | 1 | 2.8 |
| $D_{m\text{-}ENGA}$ | 1.1 | 2 | 0.98 | 1 |

# 1. Conclusion

In this report, we examined the COAs planning problem. We suggest a spatio-temporal graph oriented model to represent a COA. COA planning is then seen as decomposing a complex mission into a set of tasks or activities, with interdependence relationships and required combinations of resources or capabilities. Scheduling these activities and assigning resources or capabilities to them is a very complex hard problem. In order to solve the problem, we proposed to explore the evolutionary algorithms. In particular, this report presents a comprehensive investigation of multi-objective genetic algorithms to solve the COA planning problem.

The first part of this work has focused on the benefits of evolutionary algorithms to optimize multi-objective combinatorial problems identified as NP-Hard such as COAs planning. A brief review is reported on multi-objective optimisation and the most popular evolutionary-based methods with their advantages and disadvantages. The common features that make theses algorithms powerful are highlighted, and the concepts such as elitism, niche size and fitness sharing, crowding, non-dominated solution archive are retained. The performance of an algorithm is evaluated according to its convergence, i.e. the proximity of its outcome to the true Pareto front, and the diversity of the non-dominated generated solutions. A review on metrics for performance assessment is also presented. The completeness and the compatibility are the two properties (necessary and sufficient conditions) involved in a quality metric to state the dominance relation between two algorithms outcomes and consequently the quantitative comparison of their performance. Most of quality metrics published in the literature have not these properties and can only state about the proximity or the diversity except three binary performance indicators (Zitzler et al., 2003).

In the second part of the work, a new efficient multicriteria filtering genetic algorithm, MFGA, is proposed to optimize a highly resource-constrained project scheduling problem such as COA planning. It integrates multicriteria decision aid methods in addition to the dominance concept to achieve good diversity and proximity of the approximation set. The convergence is ensured by the dominance-based selection which is improved by multicriteria filtering method. This new operator selects efficient solutions characterized by at least one best scored objective or by all objectives achieving minimal threshold values. The diversity is achieved by two mechanisms:

– an enlarged sampling size scheme

– a candidate selection procedure based on roulette wheel mechanism, for recombination.

The proposed approach is not restricted by the features of the objective functions. Fuzzy, nominal, cardinal non linear functions can be considered. The formulation of the fitness as a vector of functions considering both objectives and constraints allows the search to be enforced towards the Pareto front from both the feasible and unfeasible regions. For the larger problem size tested in this work (COAs with mote than 50 tasks) it is found that MFGA outperforms efficient MOEAs of the literature such as ENGA, NSGAII, and SPEA2. Moreover, MFGA provides solutions sets with a higher proximity and coverage in objective space. However, its weakness is its CPU time consuming which is higher compared to NSGAII and SPEA2. Reducing the run-time complexity by using the methods proposed in computer science is one of the work's perspectives.

# 2. References


[1] Bäck, T. (1996), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York.

[2] Bäck, T., D. Fogel, Z. Michalewicz (1997), *Handbook of Evolutionary Computation*, vol.1, IOP Publishing Ltd. And Oxford University Press.

[3] Baita, F., F. Mason, C. Poloni, W. Ukovich (1995), Genetic algorithms with redundancies for the vehicle scheduling problem, In: Evolutionary Algorithms in Management Applications, J. Biethahn, V. Nissen Eds., pages 341-353, Springer-Verlag, Berlin.

[4] Cantú-Paz, E. (1997), A survey of parallel genetic algorithms. *Technical Report 97003*, Illinois Genetic Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois.

[5] Cochran, J.K., S.-M. Horng, J.W. Fowler (2000), A multi-population genetic algorithmto solve multi-objective scheduling problems for parallel machines, Working Paper. (submitted to *Computers and Operations Rsearch*).

[6] Coello Coello, C.A. (1996), *An Empirical Study of Evolutionary Techniques for Multi-objective Optimization in Engineering Design*, PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA.

[7] Coello Coello, C.A. (1999), A comprehensive survey of evolutionary based multi-objective optimization techniques, *Knowledge and Information Systems*, Vol. 1(3), pages 269-308.

[8] Coello Coello, C.A., A. Hernadez Aguirre, P.B. Buckles (2002), Evolutionary multi-objective design of combinatorial logic circuits, In*: Proceeding of 2nd NASA/DoD workshop on Evolvable Hardware*, J. Lohn et al. Eds., pages 161-170, Los Alamitos, California, IEEE Computer Society.

[9] Coello Coello, C.A., A.D. Christiansen (1998), Two new GA-based methods for multi-objective optimization, *Civil Engineering Systems*, vol. 15(3), pages 207-243.

[10] Coello Coello, C.A., D.A. Van Veldhuizen, G.B. Lamont (2002), *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic Publishers.

[11] Coello Coello, C.A., G. Toscano Pulido (2001a), A micro-genetic Algorithm for multi-objective optimization, In: *First International Conference on Evolutionary Multi-Criterion Optimization*, Zitzler et al. Eds., pages 126-140, , Springer-Verlag, Lecture Notes in Computer Science No. 1993.

[12] Coello Coello, C.A., G. Toscano Pulido (2001b), Multi-objective optimization using a micro-genetic algorithm, In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO' 2001)*, Pages 274-282, San Francisco, California, Morgan Kaufmann Publishers.

[13] Cohon, J.L., D.H. Marks (1975), A review and evaluation of multi-objective programming techniques, *Water Resources Research*, vol. 11(2), pages 208-220.

[14] Corne D.W., J.D. Knowles, M.J. Oates (2000), The Pareto enveloped-based selection algorithm for multi-objective optimization, In: *Proceedings of the Parallel Problem Solving from Nature IV Conference*, M. Schoenauer, K.Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, H.-P. Schwefel Eds., pages 839-848, Paris, France, Springer. Lecture Notes in Computer Science No. 1917.

[15] Corne D.W., N.R. Jerram, J.D. Knowles, M.J. Oates (2001), PESA-II: egion-based selection in evolutionary multiobjective optimization, In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, L. Spector et al. Eds., pages 283-290, San Francisco, California, Morgan Kaufmann Publishers.

[16] Crossley, W.A., A.M. Cook, D.W. Fanjoy, V.B. Venkayya, Using the two-branch tournament genetic algorithms for multi-objective design, In: *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and materials Conference*, Long Beach, California, AIAA Paper 98-1914.

[17] Cvetković, D., I. Parmee (1999), Use of preferences for GA-based multi-objective optimization, In: *Genetic and Evolutionary computation Conference (GECCO'99)*, W. Banzhaf et al. Eds., vol. 2, pages 1504-1509, Orlando, Florida, USA, Morgan Kaufmann Publishers.

[18] Das, I., J. Dennis (1997), A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems, *Structural Optimization*, vol. 14(1), pages 63-69.

[19] De Jong (1975), A., *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD Thesis, University of Michigan, USA.

[20] Deb, K. (1991), *Binary and Floating-Point Function Optimization using Messy Genetic Algorithms*, PhD thesis, University of Alabama, Tuscaloosa, Alabama.

[21] Deb, K. (1998), Multi-objective genetic algorithms, Problem difficulties and construction of test problems. *Technical Report CI-49/98*, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany.

[22] Deb, K. (1999a), Solving goal programming problems using multi-objective genetic algorithms, In: 1999 *Congress on Evolutionary Computation*, pages 77-84, Washington D.C., IEEE Service Center.

[23] Deb, K. (1999b), Non-linear programming using multi-objective genetic algorithms, *Technical Report CI-60/98*, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany.

[24] Deb, K. (2000a), An efficient constraint handling method for genetic algorithm, *Computer Methods in Applied Mechanics and Engineering*, vol. 186(2/4), pages 311-338.

[25] Deb, K. (2000b), Multi-objective evolutionary optimization: Past, present, and future. In: *Proceedings of the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2000)*, pages 225-236, PEDC, University of Plymouth, UK, Springer, London.

[26] Deb, K., D.E. Goldberg (1989), An investigation of niche and species formation in genetic function of optimization, in *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan-Kaufmann Publishers, pages 97-106.

[27] Dhingra, A.K., B.H. Lee (1994), A genetic algorithm approach to single and multi-objective structural optimization with discrete-continuous variables, *International Journal for Numerical Methods in Engineering*, vol. 37, pages 4059-4080.

[28] Duarte, N., A.E. Ruano, C. Fonseca, P. Flemming (2000), Accelerating multi-objective control system design using neuro-genetic approach, In: *2000 Congress on Evolutionary Computation*, vol. 1, pages 392-397, Piscataway, New-Jersey, IEEE Service Center.

[29] Edgeworth F.Y. (1881), *Mathematical Physics*, P. Keagan , London, England.

[30] Erickson, M., . Mayer, J. Horn (2001), The niched Pareto genetic algorithm 2 applied to the design of groundwater remediation systems, *First International Conference on Evolutionary Multicriterion Optimization*, Zitzler et al. Eds., pages 685-695, , Springer-Verlag, Lecture Notes in Computer Science No. 1993.

[31] Esbensen, H., E.S. Kuh, (1996), Design space exploration using the genetic algorithm, *IEEE International Symposium on Circuits and Systems (ISCAS'96)*, pages 500-503, Piscataway, NJ, IEEE.

[32] Fogel, D.B., A. Ghozeil (1997), A note on representations and variation operators, *IEEE Transaction Evolutionary Algorithms*, vol.1 (2), pages 159-161.

[33] Fonseca, C.M., P.J. Flemming (1993), Genetic algorithm for multi-objective optimization: Formulation, discussion and generalization, *Fifth International Conference on Genetic Algorithms*, pages 419-423, San-Mateo, California.

[34] Fonseca, C.M., P.J. Flemming (1995), An overview of evolutionary algorithms in multi-objective optimization, *Evolutionary Computation*, vol. 3(1), pages 1-16.

[35] Fonseca, C.M., P.J. Flemming (1998), Multi-objective optimization and multiple constraint handling with evolutionary algorithms-Part I: A unified formulation, *IEEE Transactions on Systems, Man and Cybernetics*, *Part A: Systems and Humans*, vol. 28(1), pages 26-37.

[36] Fourmen, M.P. (1985), Compaction of symbolic layout using genetic algorithms, In: *International Conference on Genetic Algorithms and their Applications*, J.J. Grefenstette (Ed.), Pittsburgh, pages 141-153.

[37] Gandibleux, X., H. Morita, N. Katoh (2001), The supported solutions used as a genetic information in a population heuristic, In : In: *First International Conference on Evolutionary Multi-Criterion Optimization*, E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne Eds., pages 429-442, Springer-Verlag. Lecture Notes in computer science No.1993.

[38] Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, M.A.

[39] Goldberg, D.E., J. Richardson (1987), Genetic algorithm with sharing for multimodal function optimization, In: *Genetic Algorithms and their Applications*, J. Grefenstette Ed., pages 41-49, Hillsdale, NewJersey, Lawrence Erlbaum.

[40] Goldberg, D.E., K. Deb, H. Kargupta, G. Harik (1993), Rapid, accurate optimization of difficult problems using fast messy genetic algorithms, In: *Proceedings of the Fifth International Conference on genetic Algorithms*, S. Forrest Ed., pages 56-64, San Mateo, CA.

[41] Greenwood, G., X.S. Hu, J.G. D'Ambrosio (1997), Fitness functions for multiple objective optimization Preoblems: Combining preferences with Pareto ranking, In: *Foundations of Genetic Algorithms 4*, R.K. Belew, M.D. Vose Eds., pages 437-455, Morgan Kaufmann, San Mateo, California.

[42] Hajela, P., C.Y. Lin (1992), Genetic search strategies in multicriterion optimal design, *Structural Optimization*, vol. 4, pages 99-107.

[43] Hertz, A., D. Klober, A framework for the description of evolutionary algorithms, *European Journal of Operational Research*, vol.26 (1), pages 1-12.

[44] Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and artificial Intelligence*, MIT press, Cambridge, MA.

[45] Horn, J. (1997), Multicriteria decision making, In: *Handbook of Evolutionary Computation*, T. Bäck, D. Fogel, Z. Michalewicz Eds, pages F1.9:1-F1.9:15, IOP Publishing Ltd. And Oxford University Press.

[46] Horn, J., N. Nafpliotis (1993), Multi-objective optimization using the niched Pareto genetic algorithm, *Technical Report IlliGal Report 93005*, University of Illinois at Urbana-Champaign, Illinois, USA.

[47] Horn, J., N. Nafpliotis, D.E. Goldberg (1993), A niched Pareto genetic algorithm for multi-objective optimization, In: *Proceedings of the first IEEE Conference on Evolutonary Computation, IEEE world Congress on Computational Intelligence*, vol. 1, pages 82-87,, Piscataway, N.J. IEEE Service Center.

[48] Hu, X.S., G. Greenwood, J.G. D'Ambrosio (1996), An evolutionary approach to harware/software partitioning, In: *Parallel Problem Solving from Nature-PPSN IV*, H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel Eds., pages 900-909, Sringer-Verlag. Lecture Notes in Computer Science No. 1141.

[49] Hwang, C.-L., A.S.M. Masud (1979), Multiple objective decision making methods and applications, In: *Lecture Notes in Economics and Mathematical Systems*, vol. 164, Springer-Verlag, New York.

[50] Ishibuchi, H., T. Murata (1998), Multi-objective genetic local search and its application to flowshop scheduling, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8(3), pages 392-403.

[51] Jaszkiewicz, A. (2000), On the performance of multiple objective genetic local search on the 0/1 knapsack problem. A comparative experiment. *Technical Report RA-002/2000*, Institute of Computer science Poznań University of Technology, Poznań, Poland.

[52] Jaszkiewicz, A. (2001), *Multiple Objective Metaheuristic Algorithms for Combinatorial optimization*, Habilitation thesis, Poznań University of Technology, Poznań, Poland.

[53] Jaszkiewicz, A., M. Hapke, P. Kominek (2001), Performance of multiple objective evolutionary algorithms on a distribution system design problem-Computational experiment. In: *First International Conference on Evolutionary Multi-Criterion Optimization*, E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne Eds., pages 241-255, Springer-Verlag. Lecture Notes in computer science No.1993.

[54] Jones, B.R., W.A. Crossley, A.S. Lyrintzis (1998), Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm, In: *Proceeding of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 98-4811.

[55] Keeney, R.L., H. Raiffa (1976), *Decisions with Multiple Objectives: preferences and Value Trade-offs*, John Wiley & Sons, New York.

[56] Kinnear, K.E., R.E. Smith, Z. Michalewicz (1997), Derivates methods, In: *Handbook of Evolutionary Computation*, T. Bäck, D. Fogel, Z. Michalewicz Eds, pages B1.5:1-B1.5:15, IOP Publishing Ltd. And Oxford University Press.

[57] Kita, H., Y. Yabumoto, N. Mori, Y. Nishikawa (1996), Multi-objective optimization by means of the thermodynamical genetic algorithm, In: *Parrallel Problem Solving from nature-PPSN IV*, Voigt et al. Eds., pages 504-512, Springer-Verlag. Lecture Notes in Computer Science No. 1141, Berlin, Germany.

[58] Knowles, J. and D.W. Corne (2000a), M-PAES: A memetic algorithm for multi-objective optimization , In: 2000 Congress on Evolutionary Computation, vol.1, pages 325-332, Piscataway, New Jersey, IEEE Service Center.

[59] Knowles, J. and D.W. Corne (2000b), Approximating the non-dominated front using the Pareto archived evolutionary strategy, *Evolutionary Computation*, vol. 8(2), pages 149-172.

[60] Laumanns, M., E. Zitzler, L. Thiele (2001), On the effects of Archiving, elitism, and density based selection in evolutionary multi-objective optimization, *First International Conference on Evolutionary Multi-Criterion Optimization*, E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne, pages 181-196, Springer-Verlag. Lecture Notes in Computer Science No. 1993.

[61] Lis, J. and A.E. Eiben (1996), A multi-sexual genetic algorithm for multi-objective optimization, In: *Proceedings of the 1996 International Conference on Evolutionary Computation*, pages 59-64, Nagoya, Japan, IEEE.

[62] Loughlin, D.H., S. Ranjithan (1997), The neighborhood constraint method: A genetic algorithm based multi-objective optimization technique. In: *Proceedings of the 7th International Conference on Genetic Algorithms*, T. Bäck Ed., pages 666-673, San Mateo, California, Michigan State University, Morgan Kaufman Publishers.

[63] Mahfoud M.W., D.E. Goldberg (1995), Parallel recombinative simulated annealing: a genetic algorithm, *Parallel Computing*, vol. 21, pages 45-52.

[64] Mitchell, M. (1996), *An introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts.

[65] Mori, N., J. Yoshida, H. Tamaki, H. Kita, Y. Nishikawa (1995), Thermodynamical selection rule for the genetic algorithm, In: *Proceedings of the second IEEE Conference on Evolutionary Algorithms*, D.B Fogel Ed., pages 188-192, Piscataway, New Jersey, IEEE Service Center.

[66] Norris, S.R., W.A. Crossley (1998), Pareto-optimal controller gains generated by a genetic algorithms , In: *AIAA 36th Aerospace Science meeting and Exhibit*, Reno, Nevada, AIAA Paper 98-0010.

[67] Oei, C.K., D.E. Goldberg, S.-J. Chang (1991), Tournament selection, niching, and the preservation of diversity, *Technical Report 91011*, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Illinois, USA.

[68] Osyczka, A. (1985), Multi-criteria optimization for engineering design, In: *Design Optimization*, J.S Gero Ed, pages 193-227, academic Press.

[69] Osyczka, A. and S. Kundu (1995), A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm, *Structural Optimization*, vol. 10(2), pages 94-99.

[70] Osyczka, A. and S. Kundu (1996), A modified distance method for multicriteria optimization using genetic algorithms, *Computers in industrial Engineering, Special Issue on genetic Algorithms*, vol. 30(40), pages 871-882.

[71] Pareto, V. (1896), *Cours d'Économie Politique*, volume I and II, F.Rouge, Lausanne.

[72] Purshouse, R.C., P.J. Flemming (2001), The multi-objective genetic algorithm applied to benchmark problems- An analysis, *Technical Report No. 796*, Department of Automatic Control and Systems Engineering, University of Sheffield, UK.

[73] Rao, S.S. (1993), Genetic algorithm approach for multi-objective optimization of structures, In: *ASME Annual Winter Meeting, Structures and Controls Optimization*, volume AD-vol.38, pages 29-38, New Orleans, Louisiana, ASME.

[74] Rekiek, B. (2000), *Assembly Line Design (Multiple Objective Grouping Genetic Algorithm and the Balancing of Mixed-Model Hybrid Assembly Line)*, PhD thesis, Free University of Brussels, CAD/CAM department, Brussels, Belgium.

[75] Richardson, J.T., M.R. Palmer, G. Liepins, M. Hilliard (1989), Some guidelines for genetic algorithms with penalty functions, In: *Proceeding of the 3rd International Conference on Genetic Algorithms*, J.D. Schaffer Ed., Pages 191-197, George Mason University, Morgan Kaufmann Publishers.

**[76]** Rowe, J., K. Vinsen, N. Marvin (1996), Parallel Gas for multiobjective functions, In: *Proceedings of the Second Nordic Workshop on Genetic Algorithms and Their Applications (2NWGA)*, J.T. Alender Ed., pages 61-70, Vaasa, Finland, University of Vaasa.

[77] Sandgren, E. (1994), Multicriteria design optimization by goal programming, In: *Advances in Design Optimization*, H. Adeli Ed., pages 225-265, Chapman & Hall, London.

[78] Schaffer J.D. (1985), Multiple objective optimization with vector evaluated genetic algorithm, In: *Genetic Algorithm and their Applications: Proceeding of the First International Conference on Genetic Algorithms*, J.J. Grefenstette Ed., pages 93-100.

[79] Sefrioui, M., J. Periaux (2000), Nash genetic algorithm: examples and applications, In: *2000 Congress on Evolutionary Computation*, vol.1, pages 509-516, Piscataway, NJ., IEEE Service Center.

[80] Srigiriraju, K.C. (2000), *Noninferior Surface Tracing Evolutionary Algorithm (NSTEA) for multiobjective optimization*, Master's thesis, North Carolina State University, Raleigh, North Carolina.

[81] Srinivas, N. and K. Deb (1995), Multiobjective optimization using non-dominated sorting in genetic algorithms, *Evolutionary Computation*, vol. 2 (3), pages 221-248.

[82] Steuer, R.E. (1986), *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York.

[83] Surry, P.D., N.J. Radcliffe, I.D. Boyd (1995), A multi-objective approach to constrained optimization of gas supply networks: the COMOGA method. In: *Evolutionary Computing. AISB Workshop. Selected Papers*, T.C. Forgaty Ed., pages 166-180, Springer-Verlag. Lecture Notes in Computer Science No. 993, Shieffeld, U.K.

[84] Tabucanon M.T. (1989), *Multiple Criteria Decision Making in Industry*, Elsevier Science Publishers.

[85] Tamaki, H., H. Kita, S. Kobayashi 91996), Multi-objective optimization by genetic algorithms: A review, In: *Proceedings of the 1996 International Conference on Evolutionary Computation (ICEC'96)*, T. Fukuda and T. Furuhashi Eds., Pages 517-522, Nagoya, Japan, IEEE.

[86] Van Veldhuizen, D.A. (1999), *Multiobjective evolutionary Algorithms: Classifications, Analysis, and New Innovations*, PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

[87] Van Veldhuizen, D.A., G.B. Lamont (2000), multiobjective optimization with messy genetic algorithm, In: *Proceedings of the 2000 ACM Symposium on Applied Computing*, pages 470-476, Villa Olmo, Como, Italy, ACM.

[88] Voget, S., M. Kolonko (1998), multidimensional optimization with a fuzzy genetic algorithm, *Journal of Heuristics*, vol 4(3), pages 221-244.

[89] Wang J.F., J. Periaux (2001), Multi-point optimization using GAs and Nash/Stackelberg Games for high lift multi-airfoil design in aerodynamics, In: *2001 Congress on Evolutionary Computation*, vol.1, pages 552-559, Piscataway, New Jersey, IEEE Service Center.

[90] Wienke, P.B., C. Lucasius, G. Katerman (1992), Multicriteria target optimization of analytical procedures using genetic algorithms, *Analytical Chimica Acta*, vol. 265(2), pages 211-225.

[91] Wilson, P.B., M.D. Macleod (1993), Low implementation cost IIR digital filter design using genetic algorithm, In: *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, pages 4/1-4/8, Chelmsford, U.K.

[92] Wolpert, D.H., W.G. Macready (1997), No free lunch theorems for optimization, *IEEE Transaction Evolutionary Algorithms*, vol.1 (1), pages 67-82.

[93] Zebulum, R.S., M.A. Pacheco, M. Vellasco, (1998), Synthesis of CMOS operational amplifiers through genetic algorithms, In: *Proceedings of the Brazilian Symposium on Integrated Circuits, SBCCI'98*, pages 125-128, Rio de Janeiro, Brazil, IEEE.

[94] Zhou, G., M. Gen (1997), Evolutionary Computation on multicriteria production process planning problem, In: *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, W. Porto Ed., pages 419-424, Piscataway, New Jersey, IEEE Press.

[95] Zhou, G., M. Gen (1999), Genetic algorithm approach on multi-criteria minimum spanning tree problem*, European Journal of Operational Research*, vol. 114 (1).

[96] Zitzler, E. and L. Thiele (1999), Multiobjective evolutionary algorithms: A comparison case study and the strength Pareto approach, *IEEE Transaction on Evolutionary Computation.*, vol 3(4), pages 257-271.

[97] Zitzler, E., M. Laumanns, L. Thiele (2001), SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, In: *EUROGEN 2001- Evolutionary Methods for Design, Optimization and Control with Applications to Industrial problems*, Barcelona, Spain.

[98] Zydallis, J.B., D.A. Van Veldhuizen, G.B. Lamont (2001), A statistical comparison of multiobjective evolutionary algorithms including the MOMGAII, In: *First International Conference on Evolutionary Multi-Criterion Optimization*, E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne Eds., pages 226-240, Springer-Verlag. Lecture Notes in computer science No.1993.

This page intentionally left blank.

# List of symbols/abbreviations/acronyms/initialisms

| | |
|---|---|
| DND | Department of National Defence |
| DRDC | Defence Research & Development Canada |
| DRDKIM | Director Research and Development Knowledge and Information Management |
| R&D | Research & Development |

# Distribution list

Document No.: DRDC Valcartier TR 2003-372

**LIST PART 1: Internal Distribution by Centre**

1 Director General
1 Chief Scientist
3 Document Library
1 Head C2 Decision Support System Section
1 Head Intelligence and Information Section
1 Head Systems of Systems Section
1 A. Auger
1 A. Benaskeur
1 A. Boukhtouta
1 A. Guitouni (Author)
1 A. Sahi
1 A.-C. Boury-Brisset
1 A.-L. Jousselme
1 C. Daigle
1 D. Gouin
1 F. Rhéaume
1 G. Thibault
1 H. Irandoust
1 J. Berger
1 J. Roy
1 Lcdr E. Woodliffe
1 Lcol M. Gareau
1 M. Allouche
1 M. Bélanger
1 M. Blanchette
1 Maj A. Lamontagne
1 N. Pageau
1 P. Valin
1 P. Maupin
1 R. Breton
1 R. Lecocq
1 S. Paradis

34 TOTAL LIST PART 1

**LIST PART 2: External Distribution by DRDKIM**

1 Library and Archives Canada
1 Director Research and Development Knowledge and Information Management (PDF file)

2 Director CORA
101 Colonel By Drive, Ottawa, ON, K1A0K2
2 Director Science and Technology (C4ISR)
Constitution Building, 305 Rideau St., Ottawa, ON, K1N 9E5
2 Director Science and Technology (Air)
Constitution Building, 305 Rideau St., Ottawa, ON, K1N 9E5
2 Director Science and Technology (Land)
Constitution Building, 305 Rideau St., Ottawa, ON, K1N 9E5
2 Director Science and Technology (Maritime)
Constitution Building, 305 Rideau St., Ottawa, ON, K1N 9E5
2 Director Science and Technology (Human Performance)
Constitution Building, 305 Rideau St., Ottawa, ON, K1N 9E5

14 TOTAL LIST PART 2

**48 TOTAL COPIES REQUIRED**

This page intentionally left blank.

## DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Comparison and Evaluation of Multi-Objective Genetic Algorithms for Military Planning and Scheduling Problems: Applied to Course of Action Planning

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Guitouni, A.; Belfares, L.

5. DATE OF PUBLICATION (Month and year of publication of document.)

April 2008

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

84

6b. NO. OF REFS (Total cited in document.)

98

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

13dm

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Valcartier TR 2003-372

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

Planning detailed military Courses of Action (COAs) is a very complex and time consuming. In this work, we investigate Evolutionary algorithms (EAs) to solve COAs'resource management and scheduling problems. The performance of such algorithms is assessed based on their outcomes quality. Efficient algorithms exhibit a good approximation of the Pareto optimal sets while requiring reasonable computational resources. A good approximation set is the resultant of a trade-off between diversity of solutions and their proximity to the true Pareto front. Such a balance is difficult to achieve with NP-hard problems exhibiting Pareto frontier discontinuity and multimodality although EAs are able to handle such optimization features. In this report, the multicriteria filtering genetic algorithm (MFGA) is proposed to achieve balanced proximity-diversity of the generated solutions. It uses a reproduction procedure based on a multicriteria filtering method and dominance concept to select solutions characterized by at least one best-scored objective or by all objectives achieving minimal threshold values. It is applied together with crossover and mutation operators. We illustrate this multi-objective EA in an enlarged sampling size scheme to solve highly constrained course of action planning problems. Cardinal and ordinal objectives are considered. An empirical comparison with three state-of-the-art multi-objective EAs is done using metrics of performance. The results show that this new approach is highly competitive, especially when applied to high dimensional problems such as those frequently encountered in real world applications.

La planification de suites d'actions militaires (COAs) est très complexe et demande beaucoup de temps. Dans ce travail, nous examinons des algorithmes évolutionnaires (AE) pour résoudre la gestion de ressources et des problèmes d'ordonnancement d'une COA. Les AEs sont de plus en plus utilisés pour résoudre des problèmes d'optimization. La performance de tels algorithmes est évaluée sur la base la qualité des solutions générées. Les algorithmes sont efficaces s'ils sont capables de trouver de bonnes approximations des solutions Pareto optimales tout en exigeant des ressources informatiques raisonnables. Un bon ensemble de solutions est celui qui permet de trouver un compromis entre la diversité des solutions et leur proximité de la frontière Pareto. Un tel équilibre est difficile à réaliser avec des problèmes NP-Durs. Dans le cadre de ce rapport, nous proposons un algorithme génétique basé sur le filtrage multicritère (MFGA) pour générer des solutions de cette qualité. MFGA utilise une procédure de reproduction basée sur une méthode de filtrage multicritère et le concept de dominance pour choisir des solutions caractérisées par la meilleure évaluation sur un des objectifs ou des évaluations supérieures à un certain seuil pour tous les objectifs. Des objectifs cardinaux et ordinaux peuvent être considérés. Cette procédure est appliquée séquentiellement aux opérateurs de croisement et de mutation. Une comparaison empirique de l'algorithme proposé avec trois AEs de la littérature est faite en se basant sur des métriques de performance. Les résultats montrent que cette nouvelle approche est très compétitive particulièrement pour des problèmes de grandes tailles, ce qui est le cas des problèmes de planification militaire.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Genetic Algorithms; Multiple Objectives Optimization; Multiple Criteria Analysis; Measurement of Performance; Course of Action Planning; Military Planning

**Defence R&D Canada**

Canada's Leader in Defence
and National Security
Science and Technology

**R & D pour la défense Canada**

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

**www.drdc-rddc.gc.ca**