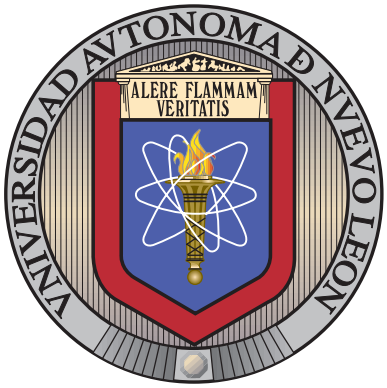
Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado



Performance evaluation and optimization of swarms of robots in a specific task

por

Luis Ángel Márquez Vega

como requisito parcial para obtener el grado de

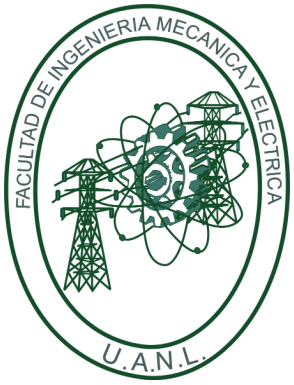
maestría en ciencias de la ingeniería eléctrica

Agosto 2019

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado



Performance evaluation and optimization of swarms of robots in a specific task

por

Luis Ángel Márquez Vega

como requisito parcial para obtener el grado de

maestría en ciencias de la ingeniería eléctrica

Agosto 2019

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Performance evaluation and optimization of swarms of robots in a specific task», realizada por el alumno Luis Ángel Márquez Vega, con número de matrícula 1521009, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ciencias de la Ingeniería Eléctrica.

El Comité de Tesis

Dr. Luis Martín Torres Treviño
Asesor

Dr. Juan Ángel Rodríguez Liñán
Revisor

Dr. Giovanni Lizárraga Lizárraga
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO
FIME

San Nicolás de los Garza, Nuevo León

Esta tesis es dedicada
A mi familia y mi novia
Gracias por todo.

This thesis is dedicated
To my family and my girlfriend
Thanks for everything.

Agradecimientos

Primero, quiero agradecer a la Universidad Autónoma de Nuevo León y a la Facultad de Ingeniería Mecánica y Eléctrica por todo el apoyo brindado durante mis estudios de maestría y por permitirme utilizar sus instalaciones durante el desarrollo de esta tesis.

Del mismo modo, estoy profundamente agradecido con el Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo financiero brindado durante mis dos años de estudios.

Quiero expresar mi agradecimiento a mi asesor de tesis, el Dr. Luis Martín Torres Treviño, por compartir sus conocimientos conmigo, por su paciencia, por sus explicaciones y por guiarme durante el desarrollo de esta tesis.

También agradezco al Dr. Juan Ángel Rodríguez Liñán y al Dr. Giovanni Lizárraga Lizárraga, revisores de esta tesis, por compartir conmigo su experiencia, por sus valiosas recomendaciones y sugerencias que incrementaron la calidad de esta tesis y por ayudarme a hacer un mejor trabajo.

Quiero extender mi gratitud a todos mis profesores quienes compartieron conmigo sus conocimientos en sus respectivas áreas durante mis estudios de maestría. También agradezco a todos los profesores del programa de posgrado en Ingeniería Eléctrica quienes se tomaron el tiempo para compartir sus opiniones sobre mi trabajo de tesis.

Agradezco a mis padres, Ladislao Márquez Hernández y Aracely Vega López, su perseverancia y dedicación me han servido como modelo de vida; gracias por su incansable apoyo, por sus incontables consejos y por todos los ánimos que me han dado para que pueda conseguir mis objetivos.

Quiero también agradecer a mis hermanas y hermano; María Aracely, Judith Vianey y Ladislao, por todas las veces que me han apoyado cuando he necesitado su ayuda y por todos los consejos que definitivamente me han ayudado a concluir mis estudios de maestría.

Agradezco a mi novia, Liliana Hernández Meléndez, por apoyarme en cada proyecto que decido iniciar, por animarme, por escuchar mis problemas y ayudarme a resolverlos; además, gracias por el apoyo brindado en la elaboración de algunas figuras en esta tesis.

Quiero agradecer a mis compañeros de clase; Lucero, Melissa, Manuel, Itzel, Julio y Diana, por su apoyo brindado durante las primeras etapas de los estudios maestría y por compartir conmigo sus opiniones sobre los problemas que se me fueron presentando durante esta tesis.

Agradezco a mis amigos; Aarón, José Armando, Juan Diego, Juan Fernando y Karen Vanessa, por su amistad, por los momentos de diversión y por sus palabras de apoyo que fueron de gran ayuda en esta etapa de mi vida.

Finalmente, agradezco a Dios por siempre estar conmigo.

Acknowledgments

First, I would like to thank to the Universidad Autónoma de Nuevo León and the Facultad de Ingeniería Mecánica y Eléctrica for all the support provided during my master's studies and for allowing me to use their facilities during the development of this thesis.

In the same way, I am deeply grateful to the National Council of Science and Technology (CONACyT) for the financial support provided during my two years of studies.

I would like to express my appreciation to my thesis advisor, Luis Martín Torres Treviño, PhD., for sharing his knowledge with me, for his patience, for his explanations and for guiding me during the development of this thesis.

I also thank to Juan Ángel Rodríguez Liñán, PhD. and Giovanni Lizárraga Lizárraga, PhD., reviewers of this thesis, for sharing with me their expertise, for their valuable recommendations and suggestions that increased the quality of this thesis and for helping me to do a better job.

I would like to extend my gratitude to all my professors who shared with me their knowledge in their respective areas during my master's studies. I also thank to all the professors of the graduate program in Electrical Engineering who took the time to share their opinions about my thesis work.

I thank my parents, Ladislao Márquez Hernández and Aracely Vega López, their perseverance and dedication have served me as life model; thank you for your tireless support, for your countless advices and for all the encouragement you have given me so that I can achieve my objectives.

I would also like to thank to my sisters and brother, María Aracely, Judith Vianey and Ladislao, for all the times they have supported me when I have needed their help and for all the advices that have definitely helped me to conclude my master's studies.

Thanks to my girlfriend, Liliana Hernández Meléndez, for supporting me in every project that I decide to start, for encouraging me, for listening to my problems and helping me to solve them; also, thanks for the support given in the elaboration of some figures in this thesis.

I would like to acknowledge my classmates; Lucero, Melissa, Manuel, Itzel, Julio and Diana, for their support during the first stages of the master's studies and for sharing with me their opinions about the problems that were presented to me during this thesis.

I thank my friends; Aarón, José Armando, Juan Diego, Juan Fernando and Karen Vanessa, for their friendship, for the moments of fun and for their words of support that were very helpful at this stage of my life.

Finally, I thank God for always stay with me.

Resumen

Luis Ángel Márquez Vega.
Universidad Autónoma de Nuevo León.
Facultad de Ingeniería Mecánica y Eléctrica.

Fecha de graduación: Agosto 2019.

Título del estudio: Performance evaluation and optimization of swarms of robots in a specific task.

Número de páginas: 144.

Candidato para obtener el grado de Maestría en Ciencias de la Ingeniería Eléctrica.

Objetivos y método de estudio: Actualmente los enjambres de robots representan una alternativa para resolver un amplio rango de tareas como búsqueda, agregación, depredador-presa, forrajeo, etc. Sin embargo, determinar qué tan bien se resuelve una tarea es un problema actual importante, asignar métricas de evaluación a las tareas realizadas por enjambres de robots es muy útil para medir el rendimiento de un enjambre particular en la resolución de la tarea. Encontrar los parámetros de control de un enjambre de robots que resuelva una tarea con el mejor rendimiento posible representa muchos beneficios como ahorro de recursos energéticos y de tiempo. El objetivo general en esta tesis es evaluar y mejorar el rendimiento de un enjambre de robots en la resolución de una tarea particular, por esta razón los siguientes objetivos específicos son propuestos: 1) Describir una tarea de flocking con búsqueda de zona objetivo y determinar las métricas de evaluación que midan la resolución de la tarea;

2) Implementar las políticas de comportamiento para un enjambre simulado de cuadricópteros; 3) Implementar técnicas de optimización multi-objetivo para encontrar los mejores conjuntos de parámetros de control del enjambre que resuelvan la tarea propuesta con el mejor rendimiento posible; 4) Comparar el rendimiento de los algoritmos de optimización multi-objetivo implementados para determinar cuál algoritmo representa la mejor opción para optimizar este tipo de tareas. Diferentes métodos para controlar enjambres de robots han sido propuestos, en esta tesis un modelo bio-inspirado basado en tendencias de repulsión (Δr), orientación (Δo) y atracción (Δa) entre especies biológicas como parvadas de aves y escuelas de peces es aplicado en el enjambre de cuadricópteros simulado. Diferentes experimentos son propuestos, la tarea de flocking con búsqueda de zona objetivo es optimizada por enjambres de cuadricópteros de 5, 10 y 20 miembros y con dos diferentes condiciones en el ambiente, un caso sin obstáculos y otro caso con obstáculos en la arena. La tarea es evaluada por cuatro funciones objetivo propuestas formuladas como problemas de minimización las cuales están orientadas a alcanzar cuatro objetivos principales en la tarea, al minimizar estas funciones objetivo el comportamiento deseado en el enjambre de cuadricópteros es alcanzado. La Optimización por Enjambre de Partículas Multi-Objetivo (MOPSO), el Algoritmo Genético de Clasificación No-dominada II usando Evolución Diferenciada (NSGA-II-DE) y el Algoritmo Evolutivo Multiobjetivo basado en Descomposición usando Evolución Diferenciada (MOEA/D-DE) son utilizados para optimizar los parámetros de control Δr, Δo y Δa para la tarea propuesta en cada experimento. La medida de Hipervolumen (HV), una métrica C modificada (Q) y el tiempo por ciclo (TPC) son las métricas seleccionadas para evaluar el rendimiento de los algoritmos de optimización multi-objetivo.

Contribuciones y conlusiones: Los resultados obtenidos muestran que las políticas de comportamiento seleccionadas producen interacciones colaborativas entre los miembros del enjambre que benefician la resolución de la

tarea. Usar técnicas de optimización multi-objetivo directamente en el simulador del enjambre de cuadricópteros produce un pequeño número de soluciones optimizadas porque el proceso de optimización es sólo adecuado con poblaciones pequeñas y con un número reducido de ciclos debido al muy alto tiempo computacional requerido. Un modelo sustituto que represente el simulador es requerido, por esa razón, un modelo estadístico basado en nodos es propuesto para estimar las funciones objetivo de un conjunto dado de parámetros de control, este modelo estadístico permite el uso de técnicas de optimización multi-objetivo con grandes poblaciones y un número mas grande de ciclos con un tiempo computacional requerido reducido. Espacios solución y frentes de Pareto estimados con suficiente información para visualizar las relaciones entre funciones objetivo en cada experimento fueron posible con el uso del modelo sustituto propuesto. Los conjuntos optimizados de parámetros de control muestran que en todos los experimentos valores pequeños en el parámetro de control Δr son preferidos y que valores grandes en los parámetros de control Δo y Δa son preferidos en este tipo de tareas. Diferentes relaciones entre parámetros de control y funciones objetivo y diferencias en los espacios solución y los frentes de Pareto estimados pueden ser observados denotando que el número de miembros y la presencia de obstáculos tiene relevancia en la selección de los parámetros de control optimizados. Los resultados de la comparación entre algoritmos de optimización multi-objetivo muestran que cuando sólo es adecuado el uso de pequeñas poblaciones y pocos ciclos el algoritmo MOPSO supera al NSGA-II-DE. Sin embargo, sin esta limitación el NSGA-II-DE es el algoritmo más balanceado pero es el más caro en términos de tiempo computacional, el algoritmo MOPSO es el mejor en términos de HV y TPC y el MOEA/D-DE es el mejor en términos de Q.

Firma del asesor: _____

Dr. Luis Martín Torres Treviño

Abstract

Luis Ángel Márquez Vega. Graduation date: August 2019.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Title of study: Performance evaluation and optimization of swarms of robots in a specific task.

Number of pages: 144. Candidate to obtain the degree of Master of Science in Electrical Engineering.

Objectives and methodology: Nowadays the swarms of robots represent an alternative to solve a wide range of tasks as search, aggregation, predator-prey, foraging, etc. However, determining how well the task is resolved is an important current problem, assign evaluation metrics to tasks performed by swarms of robots is very useful in order to measure the performance of a particular swarm in the task resolution. Find the control parameters of a swarm of robots that resolves a task with the best possible performance represents many benefits as saving of energetic resources and time. The general objective in this thesis is to evaluate and improve the performance of a swarm of robots in the resolution of a particular task, for that reason the following specific objectives are proposed: 1) To describe a flocking task with target zone search and to determine evaluation metrics that measure the task resolution; 2) To implement behavior policies for a simulated swarm of quadrotors; 3) To im-

plement multi-objective optimization techniques in order to find the best sets of control parameters of the swarm that resolve the proposed task with the best possible performance; 4) To compare the performance of the implemented multi-objective optimization algorithms in order to determine which algorithm represents the best option to optimize this type of tasks. Different methods to control swarms of robots have been proposed, in this thesis a bio-inspired model based in repulsion (Δr), orientation (Δo) and attraction (Δa) tendencies between biological species as bird flocks and schools of fish is applied in the simulated swarm of quadrotors. Different experiments are proposed, the flocking task with target zone search is optimized for swarms of quadrotors of 5, 10 and 20 members and with two different conditions in the environment, one case without obstacles and another case with obstacles in the arena. The task is evaluated by four proposed objective functions formulated as minimization problems which are oriented to reach four main objectives in the task, as these objectives functions are minimized the desired behavior of the swarm of quadrotors is reached. The Multi-Objective Particle Swarm Optimization (MOPSO), the Nondominated Sorting Genetic Algorithm II using Differential Evolution (NSGA-II-DE) and the Multiobjective Evolutionary Algorithm based on Decomposition using Differential Evolution (MOEA/D-DE) are used to optimize the control parameters Δr, Δo and Δa for the proposed task in each experiment. The Hypervolume measure (HV), a modified C-metric (Q) and the time per cycle (TPC) are the selected metrics to evaluate the performance of the multi-objective optimization algorithms.

CONTRIBUTIONS AND CONCLUSIONS: The obtained results show that the selected behavior policies produces collaborative interactions between members of the swarm that benefit the resolution of the task. Use multi-objective optimization techniques directly on the quadrotor swarm simulator produces small number of optimized solutions because the optimization process is only suitable with small populations and with a reduced number of cycles due to the

very large computational time required. A surrogate model that represents the simulator is required, for that reason, a nodes-based statistical model is proposed in order to estimate the objective functions for a given set of control parameters, this statistical model allows the use of multi-objective optimization techniques with larger populations and larger number of cycles with a reduced computation time required. Estimated solution spaces and Pareto fronts with enough information to visualize the relationships between objective functions in each experiment were possible with the use of the proposed surrogate model. The optimized sets of control parameters show that in all the experiments small values in the control parameter Δr are preferred and large values in the control parameters Δo and Δa are preferred in this type of tasks. Different relationships between control parameters and objective functions and differences in the estimated solution spaces and Pareto fronts between experiments can be observed denoting that the number of members and the presence of obstacles has relevance in the selection of the optimized control parameters. The results from the comparison between multi-objective optimization algorithms show that when only is suitable the use of small populations and few cycles the MOPSO algorithm outperforms the NSGA-II-DE. However, without this limitation the NSGA-II-DE is the most balanced algorithm but is the most expensive in terms of computational time, the MOPSO algorithm is the best in terms of HV and TPC and the MOEA/D-DE is the best in terms of Q.

Advisor signature: _____

Luis Martín Torres Treviño, PhD

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of the problem

This thesis is oriented to find the set of control parameters with which a swarm of quadrotors is able to solve a flocking task with target zone search with the best possible performance, the control parameters define the behavior of the swarm that are based on determined behavior policies. In order to find the optimal sets of control parameters, multi-objective optimization algorithms are implemented. In this thesis the flocking task with target zone search is described, the objectives to be optimized are determined, the behavior policies are defined and a comparison between multi-objective optimization algorithms is given.

1.1.1 Background

Nowadays the swarms of robots represent a good alternative to solve difficult problems [1, 2]. Several mechanisms to control a swarm of robots have been proposed [3]; however, there is not a consensus about the most proper approach. The control systems for a swarm of robots can be classified in the following three categories: (i) Multi-agent control systems, (ii) Bio-inspired, (iii) Heuristics that include hybrids.

Multi-agent systems consider schemes of master-slave, central control, consensus control, distributed control, etc. [4]. There are a high number of bio-inspired proposals [5], since the use of evolutionary computation for design controllers of swarm systems [6, 7] and behavioral-based algorithms that take inspiration from birds flocks like Particle Swarm Optimization (PSO) [8, 9, 10], from ants colonies using chemical compounds in real robots as pheromones traces [11], from bats behaviors [12] or the use of Lévy flight [13]. Our proposal uses an universal swarm algorithm so belong to this class. Heuristics that include hybrids proposals use a mixture of heuristics with bio-inspired algorithms, i.e. the use of wave algorithms for obtain collective behaviors [14], algorithms based on adaptive response threshold model [15, 16] and algorithms that combine PSO with evolutionary techniques called Robotic Darwinian PSO (RDPSO) [17] and Fractional Order RDPSO [18].

Swarm robotics is an approach to steer systems of multiple robots that mainly takes inspiration from swarm intelligence (e.g. biological systems that are only capable to perceive local information as insect colonies, bacterial colonies, bird flocks or schools of fish) [19]. The main characteristic of these systems are the collective behaviors that emerge from the interactions between members of the swarm and their interactions with the environment [20].

Self-organization is an interesting process that consists in the emergence of order in a system with the formation of global patterns only from local interactions between elements of the system [21], many examples of self-organizing systems can be seen in the natural world [22]. The emergence of collective behaviors as coordination, cooperation, deliberation and collaboration are constantly present in biological systems with self-orgaization capabilities [23].

The systems with decentralized mechanisms that show swarm intelligence features bring advantages as high robustness, scalability and flexibility that are desirable propierties in swarm systems [20, 24], for that reason, the study of bio-inspired systems that steer artificial swarms is well motivated.

Many tasks in which collective behaviors are especially advantageous can be solved using swarm robotics (e.g. aggregation, flocking, foraging, navigation, collaborative manipulation, etc). Research works have studied the previous tasks and have defined performance metrics to evaluate how well a swarm of robots completes the task [25].

Flocking refers to a coordinated movement of large number of individuals as an united group with a common objective, this behavior emerges from local interactions between individuals, the definition of flocking is inspired by the observation of biological species as bird flocks and schools of fish [26]. Reynolds [27] defines a flock as group of objects that presents a polarized and aggregate motion while avoiding collisions and proposes a distributed behavioral model known as "Boid model" based on three rules: avoid collisions, heading alignment and keep the flock together.

Many tasks are related with flocking behaviors as exploration, target search, mapping, surveillance, navigation, covering, object transportation and guiding [28]. Flocking behaviors have been obtained in swarms of robots with distance measurement and heading alignment capabilities between members of the swarm [29]. Separation, alignment and cohesion rules have been used for simulate a swarm of robots for area exploration tasks [30].

Nowadays the use of swarms of aerial robots has increased considerably due to the capability of these systems to solve many real-life problems. Many methodologies have been proposed for the control of swarms of aerial robots in order to perform a wide variety tasks as target search, tracking, surveillance or mapping [31].

Quadrotors are commonly used in applications of swarms of aerial robots due to their benefits as the capability of vertical take-off and landing, simple mechanics, good maneuverability and increased payload [32]. Quadrotors with on-board sensing capabilities have been use as flying swarms in navigation tasks [33], pattern formation and coordinated flight with an external localization system [34] and collective motion, group stabilization and coordination with a relative localization system [35].

A decentralized control scheme based on the rules proposed by Reynols [27] has been used for flocking tasks as navigation and tracking in a swarm of quadrotors [36]. Bio-inspired systems based on virtual pheromones are used in the task of target search using aerial swarms [37].

Research works on optimization of crucial parameters of behavior-based algorithms for swarms of robots have been done. Genetic algorithms have been used for the optimization of aerial robotic swarms controlled by behavior networks for the tasks of search and surveillance [38, 39]. The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [40] has been used for the optimization of a swarm of quadrotors steered by a flocking model based on repulsion, velocity alignment and obstacle avoidance in a navigation task [41]. Differential evolution has been used for optimize an aerial swarm based on stigmergic and flocking behaviors in a target search task [42].

Trianni and López-Ibáñez [43] show the advantages of use multi-objective optimization over single-objective optimization in the field of evolutionary robotics, e.g. more varied behaviors are explored during a multi-objective optimization.

Use multi-objective optimization techniques over a platform of real quadrotors are very expensive in economic and time terms, for this reason, optimization is commonly carried out on a simulator which represents the swarm of robots in the resolution of the determined task, however when the simulator has a high computational cost, the optimization is possible only with a reduced population and reduced number of cycles/generations. In order to use optimization techniques with large population and large number of cycles/generations a surrogate model which predicts the performance of a robotic swarm in the simulator with a given set of control parameters in the proposed task is needed [44], this techniques can produce Pareto fronts and solution spaces with a greater amount of information in a reduced time. Nowadays, surrogate models have been proposed to deal with expensive multi-objective evolutionary optimization problems [45, 46].

1.1.2 Objectives

1.1.2.1 General objective

To evaluate the performance of a swarm of quadrotors steered by a bio-inspired model in the resolution of a flocking task with target zone search and to improve its performance by the correct setting of its control parameters.

1.1.2.2 Specific objectives

- To describe a flocking task with target zone search, to determine the desired global behavior of the swarm during task execution and to define objective functions that measure the performance of the swarm as a minimization problem that allows to observe the performance of the swarm with different conditions in the arena and with different swarm sizes.
- To implement a bio-inspired model that steers a simulated swarm of quadrotors and allows the emergence of collaborative tendencies among members of the swarm.
- To implement multi-objective optimization techniques in order to find the best sets of control parameters of the swarm that resolve the proposed task with a very good performance in terms of the proposed objective functions and to obtain information about the relationships between objective functions.
- To compare the performance of the implemented multi-objective optimization algorithms using different metrics in order to determine which algorithm represents the best option to optimize this type of tasks.

1.1.3 Hypothesis

If objective functions are determined such that their minimization suppose a better performance of a swarm of robots in the completion of a specific task then it will be possible to apply multi-objective optimization techniques to obtain the values of the control parameters of the swarm of robots that increase its performance in the task completion. The obtained results of the optimized control parameters will give important information about the relationships between control parameters and objective functions that allows to understand how the control parameters are related with the global behavior of the swarm of robots.

1.1.4 Justification

- The use of swarms of robots that are not steered by centralized control systems and its members only have local information of the environment suppose a greater saving of economic resources and a better adaptation to hostile or unknown environments.
- Knowing the control parameters of the swarm of robots that increase the performance in the resolution of a specific task supposes a great reduction of energetic costs since it would reduce the time of execution of a task while a better global result in the task completion is reached.
- The use of multi-objective optimization techniques have not been explored for the optimization of swarms of quadrotors steered by behavior-based models.
- This knowledge could be implemented in a swarm of robots with control parameters that change dynamically depending on the conditions of the current task, this allows to perform several tasks with a very good performance.

1.1.5 Scope

This thesis has the scope to provide the sets of control parameters of a bio-inspired swarm of quadrotors with which a behavior close to the optimum is obtained in the resolution of the proposed task. The relationships between objective functions and between control parameters with objective functions will be given in order to provide an approach of the control parameters that would produce a good performance in the realization of similar tasks. A comparison will be made about the efficiency of well known multi-objective optimization algorithms.

1.1.6 Contributions

- The flocking task with target zone search is described and the objective functions that measure the task completion are determined.
- A surrogate model named nodes-based statistical model is proposed as an alternative to estimate the evaluation of expensive multi-objective optimization problems in terms of computational time.
- An extension of the C-metric that compares a multi-objective optimization algorithm with n multi-objective optimization algorithms is described.
- The optimized sets of control parameters for the proposed task with each different condition are given and the relationships between control parameters and objective functions are detected in order to give a general view of the behavior of the proposed swarm of quadrotors during the task completion.
- A comparison of multi-objective optimization algorithms is done, giving as result that the MOPSO algorithm is the best in this type of problems when the computational time is a limiting factor, however, the NSGA-II-DE is the most balanced algorithm when the computational time is not a limiting factor.

1.2 Outline of the thesis

This thesis is composed by five chapters that begin with this introduction.

The methodology is described in the Chapter 2 and is oriented to explain all the models, algorithms, techniques and metrics implemented in this thesis. First, the flocking task with target zone search is described and its objective functions are formulated. Then, all the information about the swarm of quadrotors is given, i.e. the bio-inspired model which steer the swarm is described and the control techniques to produce a stable flight in each quadrotor are explained. A surrogate model called nodes-based statistical model is proposed in order to estimate the objective functions during the optimization process. Then, a review of three well known multi-objective optimization algorithm implemented in this thesis is given. Finally, the metrics to evaluate the performance of multi-objective optimization algorithms are defined, the bootstrap methods and the multivariate normal distribution are explained.

In the Chapter 3 the experimental setup is established in order to give all the necessary information to replicate the results obtained in this thesis. In this chapter all the experiments to be performed are defined. Then, the selected parameters for the swarm of quadrotors, the nodes-based statistical model, the optimization and the bootstrap methods are showed. Finally, the selected parameters for the multi-objective optimization algorithms are given.

The Chapter 4 contains the results which are divided into sections that include individual results of each proposed experiment.

Finally, the conclusion is given in the Chapter 5. In this chapter the discussion about the obtained results is given, the contributions are emphasized and the main topics which would be studied as future work are highlighted.

Chapter 2

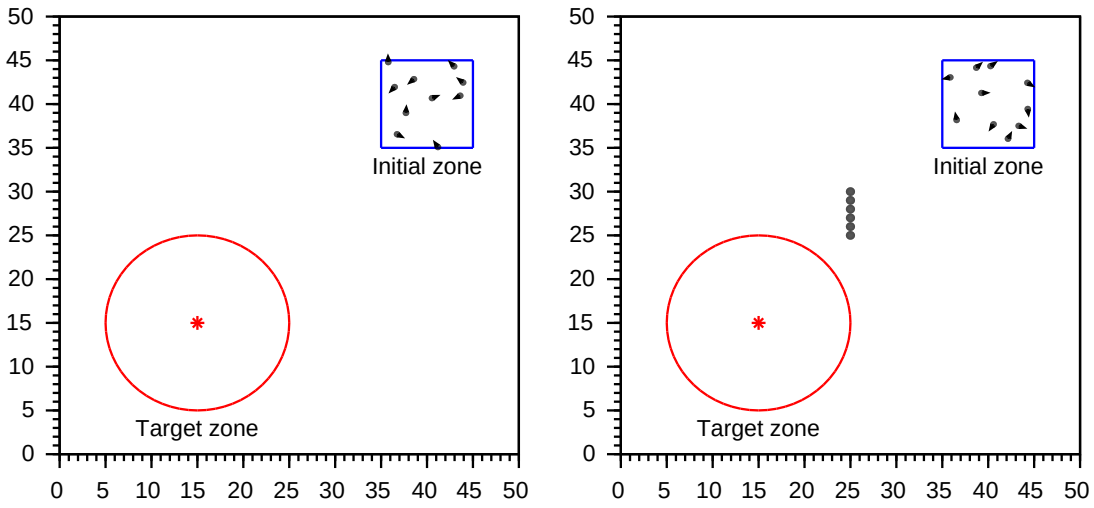
Methodology

2.1 Flocking task with target zone search

In this section the proposed flocking task with target zone search for the swarm of quadrotors is explained and the desired general behavior of the swarm is determined. The metrics that evaluate the task completion are formulated in order to measure the performance of the swarm with a given set of control parameters. The selected metrics are represented by objective functions that are selected in order to accomplish a set of general objectives in the task.

2.1.1 Description of the task

This flocking task consists in a coordinated motion of the members of a swarm in order to locate a target zone. The members of the swarm start aggregated in a square area with random positions and directions. These randomized initial conditions add noise to the task evaluation. The majority of members of the swarm must locate the target zone, this is beneficial as a stage prior to other tasks in swarms of robots as predator-prey or collaborative manipulation because as more members of the swarm stay in close proximity these tasks can be solved in a fastest way.



(a) Without obstacles (b) With obstacles

Figure 2.1: Arena of the experiment

A square arena of size 50×50 meters is selected. Two variations of the task are proposed, without and with the presence of six circular obstacles with 1 meter of diameter in the arena. The task is planned to be performed by swarms of 5, 10 and 20 members. These variations in the task must give important information about the robustness and scalability of the bio-inspired swarm of quadrotors. The localization of the initial zone of the swarm, the target zone and the obstacles is showed in the Figure 2.1

The expected behavior of the swarm in the realization of the proposed task is determined by the following general objectives:

- The target zone must be found by the swarm as fast as possible.
- All members of the swarm must stay in close proximity between them performing an aggregation task while the target zone is searched.
- Collisions with obstacles and collisions between members of the swarm must be minimum.
- The target zone must be found by the majority of members of the swarm.

2.1.2 Objective functions

In this section the objective functions which evaluate the performance of the swarm are explained. The previous general objectives of the task can be denoted by objective functions designed in order to reach an expected behavior and solve the task with the best possible performance. The mathematical formulation of every proposed objective function is shown in the following sections.

2.1.2.1 Localization time of the target zone

The execution time of a task is mainly important in swarms of quadrotors where the battery is limited, some works measure the time in which a swarm locates a target [47, 48]. In this work the average time of localization of the target zone is taken as a performance metric, in order to do that, the time of localization of the target zone by every member of the swarm is obtained and the average is calculated as follows:

$$f_1 = \frac{1}{n}\sum_{i=1}^{n} t_i \tag{2.1}$$

Where n is the total of members of the swarm, t_i is the required time for the localization of the target zone by the member i. This objective function was formulated in order to promote a collaborative behavior and penalize an individual behavior. As the localization time of every member is taken, all the members of the swarm must reach the target zone in the minimum possible time for the minimization of this objective function. If a member did not reach the target zone the reported time of localization of this member is determined by the maximum simulation time.

2.1.2.2 Average distance to the center of mass

As the flocking behavior consists in an aggregated motion of the swarm, a measure to evaluate the aggregation quality as the average distance to the center of mass is needed [6, 43]. This objective function is formulated to measure the efficiency of the bio-inspired model for maintain the members of the swarm in close proximity. The vector $\boldsymbol{c_s}$ contains the linear position of the center of mass of the swarm at the time step s and is calculated by the following equation:

$$\boldsymbol{c_s} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{r_{i,s}} \tag{2.2}$$

Where n is the total of members of the swarm and $\boldsymbol{r_{i,s}}$ is the linear position of the member i at the time step s. In order to calculate the average distance to the center of mass, the Euclidean distance between the center of mass and every member of the swarm must be calculated. As the objective is maintain the members of the swarm in close proximity during the entire realization of the task, this procedure must be done every time step during the simulation. The mathematical formulation of this objective function is shown as follows:

$$f_2 = \frac{1}{T} \sum_{s=1}^{T} \left(\frac{1}{n} \sum_{i=1}^{n} d\left(\boldsymbol{c_s}, \boldsymbol{r_{i,s}}\right) \right) \tag{2.3}$$

Where T is the total of time steps in the simulation, n is the total of members of the swarm and $d(\boldsymbol{c_s}, \boldsymbol{r_{i,s}})$ represents the Euclidean distance between the vectors $\boldsymbol{c_s}$ and $\boldsymbol{r_{i,s}}$. With the formulation of this objective function the behavior of an united swarm is better evaluated, and the formation of groups inside the swarm has a negative evaluation. This behavior promotes that if a member of the swarm finds the target zone, the others members of the swarm have a high possibility to find the target zone too.

2.1.2.3 COLLISIONS PER MEMBER

Collisions avoidance is one of the three basic rules for flocking behavior as proposed by Reynolds [27]. The minimization of collisions has been taken into account during the optimization of a navigation task performed by a swarm of quadrotors [41]. This objective function was formulated in order to minimize the number of collisions with obstacles and between members of the swarm. A matrix $\boldsymbol{D_s}$ that contains the distances between members of the swarm at time step s is formulated in order to analyze these distances and determine if collisions exist between members. The following matrix of distances is formulated:

$$\boldsymbol{D_s} = \begin{bmatrix} 0 & d_{12} & d_{13} & \dots & d_{1n} \\ d_{21} & 0 & d_{23} & \dots & d_{2n} \\ d_{31} & d_{32} & 0 & \dots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \dots & 0 \end{bmatrix} \tag{2.4}$$

The matrix $\boldsymbol{D_s}$ is a square matrix which elements of the main diagonal are 0 and the elements d_{ij} and d_{ji} are equal, the elements which are above of the main diagonal are evaluated and the total collisions in the time step s are detected. In order to obtain the relation between the number of collision and the swarm size, the total of collisions detected are divided by the number of members of the swarm as follows:

$$f_3 = \frac{1}{n} \sum_{s=1}^{T} collisions_s \tag{2.5}$$

Where n is the total of members of the swarm, T is the total of time steps in the simulation and $collisions_s$ is the total of collisions detected at time step s.

2.1.2.4 Ratio of failure

The number of members inside of a target area is a common metric in task where is preferably that all the swarm locates a target [49]. In order to obtain a minimization problem, this objective function has the purpose of determine the percentage of members of the swarm that did not reach the target zone at the end of the task. This objective function is designed in order to secure that solutions which not all the members reach the target zone have a bad evaluation. A status of position ($status_i$) is assigned to each member of the swarm as follows:

$$status_i = \begin{cases} 1, & \text{Target zone is not located} \\ 0, & \text{Otherwise} \end{cases} \tag{2.6}$$

The ratio between the members that did not reach the target zone and the total of members of the swarm is obtained by the following equation:

$$f_4 = \frac{1}{n} \sum_{i=1}^{n} status_i \tag{2.7}$$

Where n is the total of members of the swarm and $status_i$ is the status of position of member i. In this way a minimization problem is produced and the solutions with more members that reach the target zone are better evaluated. A value of 0 in this objective function means that all the members of the swarm reach the target zone, while a 1 means that the target zone was not located by any member. This objective function is very useful to determine how well the swarm accomplished the task.

2.1.3 Relationships between objective functions

The first objective function is related to the movement speed and the stigmergic collaboration of the swarm, the second and third objective functions permit an united swarm while avoiding collisions between members and the fourth objective function measure the completion of the task. The correct selection of the control parameters of the swarm is very important to minimize the previous objective functions with the best possible performance.

An important aspect in a multi-objective problem is the relationships between its objective functions. Let f_i, f_j be two objective functions and x, y be two different solutions for a multi-objective problem. Using the classification given by Purshouse [50] the following relationships are defined:

- Conflict. The objectives i and j are in a conflicting relationship if the following condition is satisfied: $f_i(x) < f_i(y) \wedge f_j(x) > f_j(y)$. If such condition is satisfied for all the possible solutions in the search space then the objectives i and j are in total conflict.

- Harmony. The objectives i and j are in a harmonious relationship if the following condition is satisfied: $f_i(x) < f_i(y) \wedge f_j(x) < f_j(y)$. If such condition is satisfied for all the possible solutions in the search space then the objectives i and j are in total harmony.

- Independence. The objectives i and j are independent if they can be separated in sub-problems and optimize independently.

The inclusion of non-conflicting objectives in a problem with multiple objectives can be beneficial, specially in problems that the relationships between objectives can change with different solutions in the search space or in real-life problems in which more knowledge about the relationships between objectives and parameters wants to be obtained [50, 51].

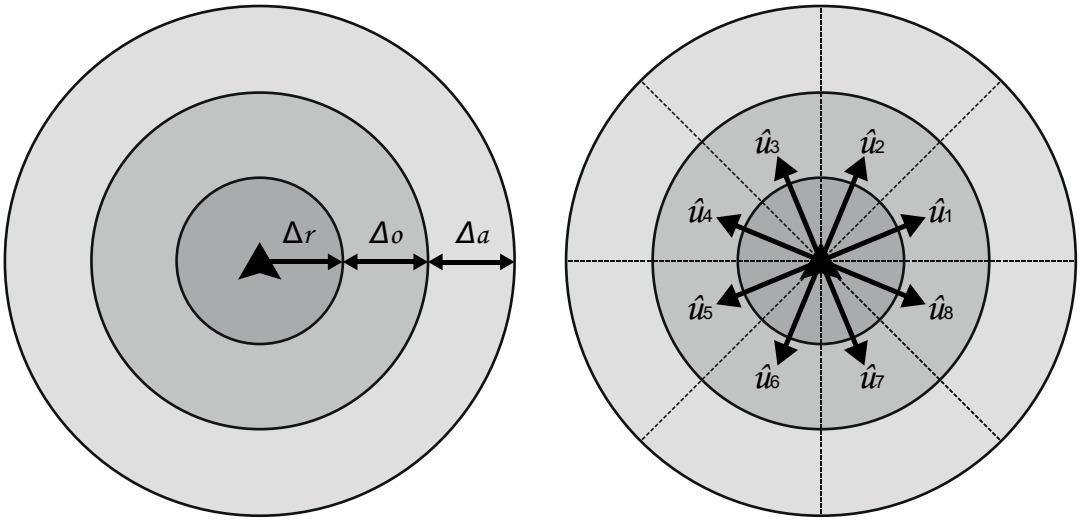
2.2 Quadrotor swarm

In this section all the corresponding information about the simulated swarm of quadrotors is given. The first part is about a bio-inspired model to control swarms, the abilities of perception and all the possible behaviors that the members of the swarm can adopt are explained. Then, the dynamic model and the control techniques implemented in each member of the swarm are explained. Finally, the algorithm that unites the behavior policies with the control techniques and represents the swarm of quadrotors in the flocking task with target zone search is represented by a pseudocode.

2.2.1 Behavior policies

The behavior of the members of the swarm is based in the model proposed by Couzin et al. [52] which uses the flocking rules proposed by Reynolds [27]. This model imitates in a more realistic way the movement of bird flocks and schools of fish and simulates the behavior of individuals based on three main characteristics of biological species; repulsion, orientation and attractive tendencies toward others individuals. As showed in the Figure 2.2a these three main characteristics are represented by zones around the individual.

The first zone is the zone of repulsion (ZOR) which is of radius Δr, the individual tries to stay away of neighbors within this zone, the second zone is the zone of orientation (ZOO) which is an annulus of width Δo, the individual tries to align in the same way of neighbors within this zone, and the last zone is the zone of attraction (ZOA) which is an annulus of width Δa, the individual tries to getting closer to neighbors within this zone. By modifying the values for the distances Δr, Δo and Δa many collective behaviors may arise between neighbors [53], this characteristic is very useful for the implementation of this model on swarms of robots.



(a) Zones (b) Sectors of view

Figure 2.2: Zones and sectors of view of each individual

The zone of influence (ZOI) which is a circle around the individual of radius Δi is added to the model in order to control the swarm, the individual tends to gets closer to the influence when it is perceived. In this proposed task a influence point is located in the center of the target zone represented by an asterisk in the Figure 2.1, the influence is perceived only when the individual is inside of the target zone.

In the proposed swarm of quadrotors every member has the capability of monitoring eight different sectors of view in order to detect neighbors, obstacles and influences. These sectors of view are represented by unit vectors as showed in the Figure 2.2b. Each unit vector $\hat{\boldsymbol{u}}_{\boldsymbol{k}}$ is formulated as:

$$\hat{\boldsymbol{u}}_{\boldsymbol{k}} = \begin{bmatrix} \cos((2k-1)\pi/8) & \sin((2k-1)\pi/8) \end{bmatrix} \tag{2.8}$$

The matrix $\boldsymbol{S_i} = [\hat{\boldsymbol{u}}_{\boldsymbol{1}}, \hat{\boldsymbol{u}}_{\boldsymbol{2}}, \ldots, \hat{\boldsymbol{u}}_{\boldsymbol{8}}]^T$ contains the unit vectors that correspond to each sector of view of the individual i.

2.2.1.1 Behaviors related with environment

Every member of the swarm has the possibility to perform different behaviors related with the environment as being attracted by influences, avoids obstacles in the arena and explores a specific direction. Each behavior is represented by an unit vector that contains the desired direction of travel due to the influences, obstacles and exploratory tendencies respectively.

In order to give to every member of the swarm the capability of detect influences, the vector $\boldsymbol{vi_i} = [vi_{i,1}, vi_{i,2}, \ldots, vi_{i,8}]$ is determined, where $vi_{i,k}$ is associated with the sector of view k of the individual i. The values of the elements of the vector $\boldsymbol{vi_i}$ are selected by using the following equation:

$$vi_{i,k} = \begin{cases} 1, & \text{Influence is detected} \\ 0, & \text{Otherwise} \end{cases} \tag{2.9}$$

The unit vector $\hat{\boldsymbol{di}}_{\boldsymbol{i}}$ that represents the desired direction of travel of the individual i due to the detected influences is given by the following equation:

$$\hat{\boldsymbol{di}}_{\boldsymbol{i}} = \frac{\boldsymbol{vi_i} \cdot \boldsymbol{S_i}}{\|\boldsymbol{vi_i} \cdot \boldsymbol{S_i}\|} \tag{2.10}$$

In order to give to every member of the swarm the capability of detect obstacles which are inside its perception range, the vector $\boldsymbol{vb_i} = [vb_{i,1}, vb_{i,2}, \ldots, vb_{i,8}]$ is determined, where $vb_{i,k}$ is associated with the sector of view k of the individual i. The values of the elements of the vector $\boldsymbol{vb_i}$ are determined by:

$$vb_{i,k} = \begin{cases} 1, & \text{Obstacle is detected} \\ 0, & \text{Otherwise} \end{cases} \tag{2.11}$$

Every member produces an unit vector with opposite direction from the detected obstacles, the unit vector $\hat{\boldsymbol{db}}_{\boldsymbol{i}}$ that represents the desired direction of travel of the individual i due to the detected obstacles is given by:

$$\hat{\boldsymbol{db}}_{\boldsymbol{i}} = -\frac{\boldsymbol{vb}_{\boldsymbol{i}} \cdot \boldsymbol{S}_{\boldsymbol{i}}}{\|\boldsymbol{vb}_{\boldsymbol{i}} \cdot \boldsymbol{S}_{\boldsymbol{i}}\|} \tag{2.12}$$

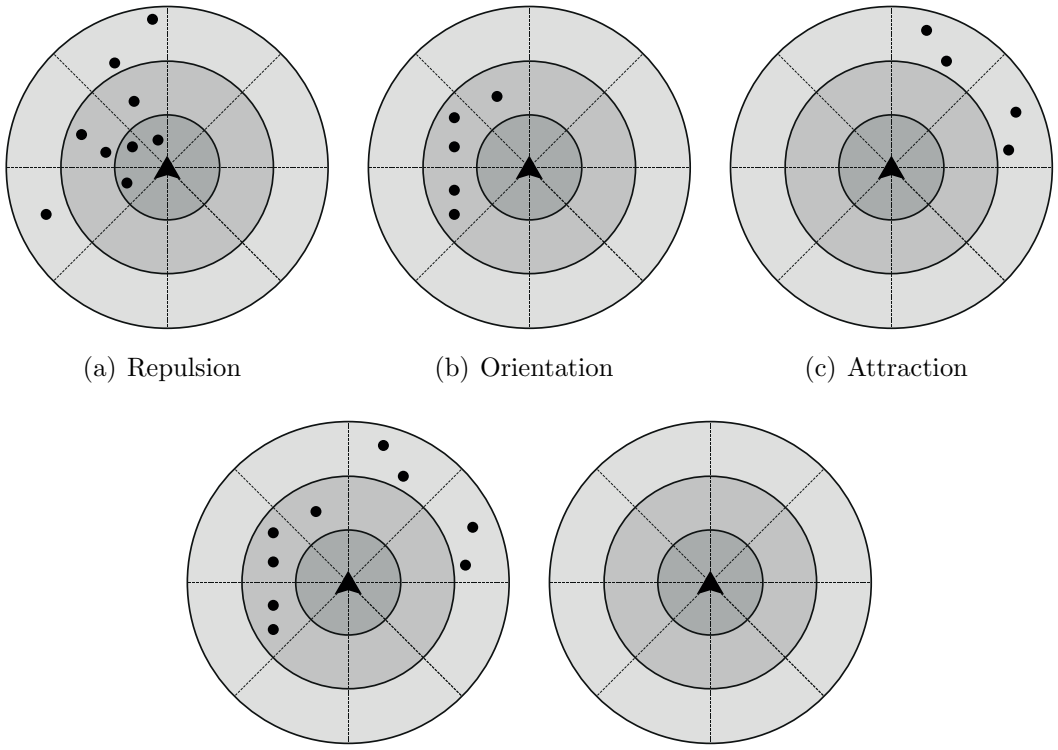
Also, every member of the swarm has a independent desired direction to explore. This desired direction is very important in order to let the member to explore an unknown area. The unit vector $\hat{\boldsymbol{de}}_{\boldsymbol{i}}$ that represents the desired direction to explore of the individual i is calculated by:

$$\hat{\boldsymbol{de}}_{\boldsymbol{i}} = \begin{bmatrix} \cos(\theta_{rand,i}) & \sin(\theta_{rand,i}) \end{bmatrix} \tag{2.13}$$

Where $\theta_{rand,i}$ is a random angle between 0 and 2π radians selected for the individual i. The angle $\theta_{rand,i}$ is modified each time that an obstacle is detected in the range of perception of the individual. This random angle is selected between 0 and π radians from the opposite semi-circle of the detected obstacle in order to explore the opposite region of the detected obstacle.

2.2.1.2 BEHAVIORS RELATED WITH NEIGHBORS

Every member of the swarm has the possibility to perform different behaviors related with the detected neighbors; gets away from neighbors in ZOR, aligns in the same way as neighbors in ZOO and gets close to neighbors in ZOA; for that reason, the selected behaviors to be performed by an individual depends on the detected neighbors in the zones of repulsion, orientation and attraction respectively. In the Figure 2.3 can be seen different examples of these behaviors. The orientation and attraction behaviors can be performed at the same time but the repulsion behavior can not be combined with other behavior related with neighbors.



(a) Repulsion (b) Orientation (c) Attraction

(d) Orientation-Attraction (e) Isolated

Figure 2.3: Examples of different behaviors related with neighbors

In order to detect neighbors in ZOR the vector $\boldsymbol{vr_i} = [vr_{i,1}, vr_{i,2}, \ldots, vr_{i,8}]$ is determined, where $vr_{i,k}$ is associated with the sector of view k of the individual i. The values of the elements of the vector $\boldsymbol{vr_i}$ are determined as follows:

$$vr_{i,k} = \begin{cases} 1, & \text{Neighbor is detected in ZOR} \\ 0, & \text{Otherwise} \end{cases} \tag{2.14}$$

The unit vector $\hat{\boldsymbol{dr_i}}$ that represents the desired direction of travel of the individual i due to the neighbors in ZOR which resulting direction is opposite from the detected neighbors is given by:

$$\hat{\boldsymbol{dr_i}} = -\frac{\boldsymbol{vr_i} \cdot \boldsymbol{S_i}}{\|\boldsymbol{vr_i} \cdot \boldsymbol{S_i}\|} \tag{2.15}$$

To every neighbor detected in ZOO a unit vector that represents its current direction of travel is formulated as follows:

$$\hat{\boldsymbol{o}}_{\boldsymbol{j}} = \begin{bmatrix} \cos(\theta_j) & \sin(\theta_j) \end{bmatrix} \tag{2.16}$$

Where θ_j is the current direction angle of the neighbor j. The unit vector $\hat{\boldsymbol{do}}_{\boldsymbol{i}}$ represents the desired direction of travel of the individual i due to the neighbors in ZOO and is given by the following equation:

$$\hat{\boldsymbol{do}}_{\boldsymbol{i}} = \frac{\sum_{j=1}^{n_o} \hat{\boldsymbol{o}}_{\boldsymbol{j}}}{\|\sum_{j=1}^{n_o} \hat{\boldsymbol{o}}_{\boldsymbol{j}}\|} \tag{2.17}$$

Where n_o is the total of neighbors detected within ZOO. In order to detect neighbors in ZOA the vector $\boldsymbol{va}_{\boldsymbol{i}} = [va_{i,1}, va_{i,2}, \ldots, va_{i,8}]$ is determined, where $va_{i,k}$ is associated with the sector of view k of the individual i. The values of the elements of the vector $\boldsymbol{va}_{\boldsymbol{i}}$ are determined as follows:

$$va_{i,k} = \begin{cases} 1, & \text{Neighbor is detected in ZOA} \\ 0, & \text{Otherwise} \end{cases} \tag{2.18}$$

The unit vector $\hat{\boldsymbol{da}}_{\boldsymbol{i}}$ that represents the desired direction of travel due to the neighbors in ZOA of the individual i which resulting direction is oriented towards the detected neighbors is given by the following equation:

$$\hat{\boldsymbol{da}}_{\boldsymbol{i}} = \frac{\boldsymbol{va}_{\boldsymbol{i}} \cdot \boldsymbol{S}_{\boldsymbol{i}}}{\|\boldsymbol{va}_{\boldsymbol{i}} \cdot \boldsymbol{S}_{\boldsymbol{i}}\|} \tag{2.19}$$

2.2.1.3 Final direction of travel

When neighbors are detected in ZOR, the final direction of travel of the individual i is given by a vector addition of the unit vectors $\hat{\boldsymbol{di}}_{\boldsymbol{i}}$, $\hat{\boldsymbol{db}}_{\boldsymbol{i}}$, $\hat{\boldsymbol{de}}_{\boldsymbol{i}}$ and $\hat{\boldsymbol{dr}}_{\boldsymbol{i}}$ as showed in the following equation:

$$\boldsymbol{d_i} = \begin{cases} w_1\hat{\boldsymbol{dr}}_{\boldsymbol{i}} + w_2\hat{\boldsymbol{db}}_{\boldsymbol{i}} + w_3\hat{\boldsymbol{de}}_{\boldsymbol{i}}, & \text{Inf. and obs. are detected} \\ w_1\hat{\boldsymbol{dr}}_{\boldsymbol{i}} + (w_2 + w_3)\hat{\boldsymbol{di}}_{\boldsymbol{i}}, & \text{Influence is detected} \\ w_1\hat{\boldsymbol{dr}}_{\boldsymbol{i}} + w_2\hat{\boldsymbol{db}}_{\boldsymbol{i}} + w_3\hat{\boldsymbol{de}}_{\boldsymbol{i}}, & \text{Obstacle is detected} \\ w_1\hat{\boldsymbol{dr}}_{\boldsymbol{i}} + (w_2 + w_3)\hat{\boldsymbol{de}}_{\boldsymbol{i}}, & \text{Otherwise} \end{cases} \tag{2.20}$$

Where the weights w_1, w_2 and w_3 are needed to give priority to the repulsion of detected neighbors. When neighbors are not detected in ZOR, the final direction of travel of the individual i is given by a vector addition of the unit vectors $\hat{\boldsymbol{di}}_{\boldsymbol{i}}$, $\hat{\boldsymbol{db}}_{\boldsymbol{i}}$, $\hat{\boldsymbol{de}}_{\boldsymbol{i}}$, $\hat{\boldsymbol{do}}_{\boldsymbol{i}}$ and $\hat{\boldsymbol{da}}_{\boldsymbol{i}}$ as follows:

$$\boldsymbol{d_i} = \begin{cases} \hat{\boldsymbol{do}}_{\boldsymbol{i}} + \hat{\boldsymbol{da}}_{\boldsymbol{i}} + \hat{\boldsymbol{db}}_{\boldsymbol{i}} + \hat{\boldsymbol{de}}_{\boldsymbol{i}}, & \text{Inf. and obs. are detected} \\ \hat{\boldsymbol{do}}_{\boldsymbol{i}} + \hat{\boldsymbol{da}}_{\boldsymbol{i}} + \hat{\boldsymbol{di}}_{\boldsymbol{i}}, & \text{Influence is detected} \\ \hat{\boldsymbol{do}}_{\boldsymbol{i}} + \hat{\boldsymbol{da}}_{\boldsymbol{i}} + \hat{\boldsymbol{db}}_{\boldsymbol{i}} + \hat{\boldsymbol{de}}_{\boldsymbol{i}}, & \text{Obstacle is detected} \\ \hat{\boldsymbol{do}}_{\boldsymbol{i}} + \hat{\boldsymbol{da}}_{\boldsymbol{i}} + \hat{\boldsymbol{de}}_{\boldsymbol{i}}, & \text{Otherwise} \end{cases} \tag{2.21}$$

In all the previous behaviors when the influence and a obstacle are detected at the same time by the individual, the influence are ignored in order to give priority to avoid collisions with the obstacle.

2.2.2 Dynamic model and control

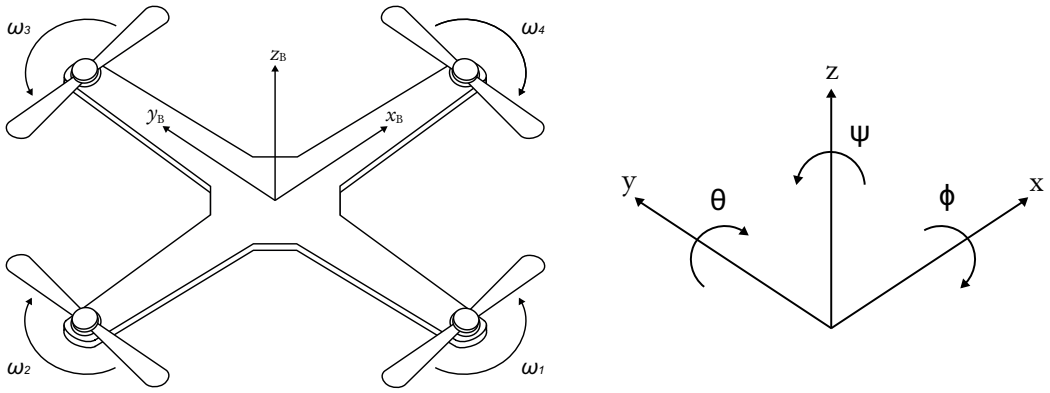
Each member of the swarm is modeled as an independent entity, the individual movement of each quadrotor is described using the same dynamic model and by using the directions of the vectors generated by the bio-inspired model that contain the final direction of travel per individual showed in the equations (2.20) and (2.21) is possible to produce a controlled flight in each quadrotor using a PID controller. Although each individual is controlled autonomously, the interactions of each individual with the environment and with adjacent neighbors produces a global and stigmergic behavior of the entire swarm as result of the selected desired directions of travel given by the bio-inspired model.

2.2.2.1 Dynamic model

The structure of the quadrotors, angular velocities w_1, w_2, w_3 and w_4 created by the four rotors and the location of the body reference frame are presented in the Figure 2.4a. The inertial reference frame which includes the absolute linear $[x, y, z]$ and the angular position $[\phi, \theta, \psi]$ are given in the Figure 2.4b, the values of the angles ϕ, θ and ψ can be determined in order to produce a desired movement in each quadrotor. The dynamic model are the same as used by Aguilera-Ruiz et al. [54].

The system that describes the linear and angular accelerations of each quadrotor is modeled by:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(c_\phi s_\theta c_\psi + s_\phi s_\psi)T - a_d\dot{x} \\ \frac{1}{m}(c_\phi s_\theta s_\psi - s_\phi c_\psi)T - a_d\dot{y} \\ -g + \frac{1}{m}(c_\phi c_\theta)T - a_d\dot{z} \\ \frac{1}{I_{xx}}\tau_\phi - \frac{I_{yy}-Izz}{I_{xx}}\dot{\theta}\dot{\psi} \\ \frac{1}{I_{yy}}\tau_\theta - \frac{I_{zz}-Ixx}{I_{yy}}\dot{\phi}\dot{\psi} \\ \frac{1}{I_{zz}}\tau_\psi - \frac{I_{xx}-Iyy}{I_{zz}}\dot{\phi}\dot{\theta} \end{bmatrix} \tag{2.22}$$



(a) Body reference frame (b) Inertial reference frame

Figure 2.4: Body and inertial reference frames

Where s_ϕ and c_ϕ represent $\sin(\phi)$ and $\cos(\phi)$ respectively, the same occurs with θ and ψ, m is the mass of the quadrotor, a_d is the air friction coefficient, g is the gravity, I_{xx}, I_{yy} and I_{zz} are the inertia in the axis x, y and z respectively. The vertical thrust T and the torques τ_ϕ, τ_θ and τ_ψ are calculated as follows:

$$\begin{aligned} T &= k(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ \tau_\phi &= lk(-\omega_1^2 + \omega_3^2) \\ \tau_\theta &= lk(-\omega_2^2 + \omega_4^2) \\ \tau_\psi &= d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{aligned} \tag{2.23}$$

Where k is the thrust coefficient, l is the arm length and d is the drag coefficient. As the dynamic model of each quadrotor has been defined, it is possible to use control techniques that allow to generate a stable and controlled flight in each member of the swarm of quadrotors, since the propeller velocities can be generated by:

$$\begin{aligned} \omega_1^2 &= \frac{u_1}{4k} - \frac{u_2}{2lk} - \frac{u_4}{4d} \\ \omega_2^2 &= \frac{u_1}{4k} - \frac{u_3}{2lk} + \frac{u_4}{4d} \\ \omega_3^2 &= \frac{u_1}{4k} + \frac{u_2}{2lk} - \frac{u_4}{4d} \\ \omega_4^2 &= \frac{u_1}{4k} + \frac{u_3}{2lk} + \frac{u_4}{4d} \end{aligned} \tag{2.24}$$

2.2.2.2 Altitude and angular position controller

The altitude and the angular positions of each member of the swarm must be controlled during the simulation, for that reason the following PID controller for the inputs u_1, u_2, u_3 and u_4 is implemented for each quadrotor [54].

$$\begin{aligned}
u_1 &= \left(g + k_p(z_d - z) + k_i \int_0^t (z_d - z)dt + k_d(\dot{z_d} - \dot{z}) \right) \frac{m}{c_\phi c_\theta} \\
u_2 &= \left(k_p(\phi_d - \phi) + k_i \int_0^t (\phi_d - \phi)dt + k_d(\dot{\phi_d} - \dot{\phi}) \right) I_{xx} \\
u_3 &= \left(k_p(\theta_d - \theta) + k_i \int_0^t (\theta_d - \theta)dt + k_d(\dot{\theta_d} - \dot{\theta}) \right) I_{yy} \\
u_4 &= \left(k_p(\psi_d - \psi) + k_i \int_0^t (\psi_d - \psi)dt + k_d(\dot{\psi_d} - \dot{\psi}) \right) I_{zz}
\end{aligned} \tag{2.25}$$

Where k_p, k_i and k_d are the proportional, integral and derivative gains respectively, ϕ_d, θ_d and ψ_d are the desired angular position of the quadrotor which are modified based on the proposed policies as result of the different cases of interaction between the quadrotor with the environment or between members of the swarm as explained in the Section 2.2.1 and z_d is the desired altitude which is kept constant through the simulation and is the same for all members of the swarm.

Using the equation (2.24) and the calculated values of u_1, u_2, u_3 and u_4 from the equation (2.25) the angular velocities w_1, w_2, w_3 and w_4 can be generated such that the vertical thrust T and the torques τ_ϕ, τ_θ and τ_ψ can be obtained from the equation (2.23) and control the system showed in the equation (2.22).

2.2.2.3 Relative linear position controller

When a member of the swarm does not detect neighbors in ZOR, the desired direction of travel and the angle ψ_d are the same, but in the opposite case a controller is required to calculate the angles ϕ_d and θ_d needed to perform the repulsion behavior.

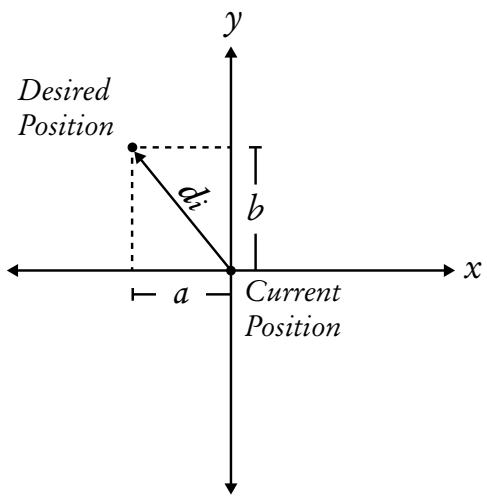


Figure 2.5: Example of desired position

The angles ϕ_d and θ_d must be calculated in order to reach the position determined by the vector $\boldsymbol{d_i}$ of the equation (2.20) from the current position as showed in the Figure 2.5. The desired angles ϕ_d and θ_d are calculated as proposed by Michael et al. [55]:

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} \frac{1}{g}(x_c s_{\psi_d} - y_c c_{\psi_d}) \\ \frac{1}{g}(x_c c_{\psi_d} + y_c s_{\psi_d}) \end{bmatrix} \tag{2.26}$$

Where the values for x_c and y_c are calculated using the PD controller:

$$\begin{aligned} x_c &= k_{P,p}(x_d - x_{rel}) + k_{P,d}(\dot{x_d} - \dot{x}) + \ddot{x_d} \\ y_c &= k_{P,p}(y_d - y_{rel}) + k_{P,d}(\dot{y_d} - \dot{y}) + \ddot{y_d} \end{aligned} \tag{2.27}$$

Where $k_{P,p}$ and $k_{P,d}$ are the proportional and derivative gains, $x_{rel} = x - x_0$ and $y_{rel} = y - y_0$, the values for x_0 and y_0 are the initial linear position. A polynomial trajectory generation [56] is used to calculate desired relative position, velocity and acceleration between two points:

$$
\begin{aligned}
x_d &= 10a(t-t_0)^3 - 15a(t-t_0)^4 + 6a(t-t_0)^5 \\
\dot{x_d} &= 30a(t-t_0)^2 - 60a(t-t_0)^3 + 30a(t-t_0)^4 \\
\ddot{x_d} &= 60a(t-t_0) - 180a(t-t_0)^2 + 120a(t-t_0)^3 \\
y_d &= 10b(t-t_0)^3 - 15b(t-t_0)^4 + 6b(t-t_0)^5 \\
\dot{y_d} &= 30b(t-t_0)^2 - 60b(t-t_0)^3 + 30b(t-t_0)^4 \\
\ddot{y_d} &= 60b(t-t_0) - 180b(t-t_0)^2 + 120b(t-t_0)^3
\end{aligned}
\tag{2.28}
$$

Where t_0 is the initial time, a and b are the x and y component of the vector $\boldsymbol{d_i}$ respectively. If the close-loop response of the system is sufficiently fast, a quadrotor performing a repulsion behavior will avoid collisions with others members of the swarm by modifying the angles ϕ and θ while maintaining its current angle ψ.

2.2.3 Simulator

As the flocking task, the bio-inspired model that steer the swarm of quadrotors, the dynamic model of quadrotors and the control techniques that allow a controlled flight of quadrotors have been defined, a simulator that represents the quadrotor swarm performing the flocking task with target zone search is needed.

The Algorithm 2.1 represents a simulator that needs as input the control parameters of the swarm of quadrotors defined in the Section 2.2.1 and gives as output the evaluation of objective functions defined in the Section 2.1.2 obtained in the flocking task with target zone search.

This simulator can be used to evaluate candidate solutions during the optimization process because it can evaluate the performance of a swarm with a given control parameters in the four proposed objective functions. However, it is very expensive in terms of computational time.

Algorithm 2.1 Quadrotor swarm simulator

Input: Δ_r, Δ_o, Δ_a;
Output: f_1, f_2, f_3, f_4;

Initialize linear and angular positions and velocities;
for $t = 0 : \Delta t : t_{max}$ **do**
  for $i = 1 :$ Total of quadrotors **do**
    Calculate distance and angle to influence;
    Update vector $\boldsymbol{vi_i}$ with equation (2.9);
    Calculate distances and angles to obstacles;
    Update vector $\boldsymbol{vb_i}$ with equation (2.11);
    if Obstacle is in the range of perception **then**
      Calculate unit vector $\hat{\boldsymbol{de_i}}$ as equation (2.13);
    end if
    for $j = 1 :$ Total of quadrotors **do**
      Calculate distance and angle to quadrotor j;
      Update vector $\boldsymbol{vr_i}$ as equation (2.14);
      Update vector $\boldsymbol{va_i}$ as equation (2.18);
      if Quadrotor j is in the ZOO **then**
        Calculate unit vector $\hat{\boldsymbol{o_j}}$ as the equation (2.16);
      end if
    end for
    Calculate unit vector $\hat{\boldsymbol{di_i}}$ as equation (2.10);
    Calculate unit vector $\hat{\boldsymbol{db_i}}$ as equation (2.12);
    Calculate unit vector $\hat{\boldsymbol{dr_i}}$ as equation (2.15);
    Calculate unit vector $\hat{\boldsymbol{do_i}}$ as equation (2.17);
    Calculate unit vector $\hat{\boldsymbol{da_i}}$ as equation (2.19);
    if If neighbors are detected in the ZOR **then**;
      Calculate $\boldsymbol{d_i}$ as the equation (2.20);
      Calculate ϕ_d and θ_d as the equation (2.26);
      Set ψ_d as current angle ψ;
    else
      Calculate $\boldsymbol{d_i}$ as the equation (2.21);
      Set ϕ_d and θ_d for a constant speed;
      Set ψ_d as direction of vector $\boldsymbol{d_i}$;
    end if
    Use the PID controller;
    Calculate acceleration, velocity and position;
  end for
  Update objective functions;
  if All quadrotors are in the target zone **then**
    break;
  end if
end for

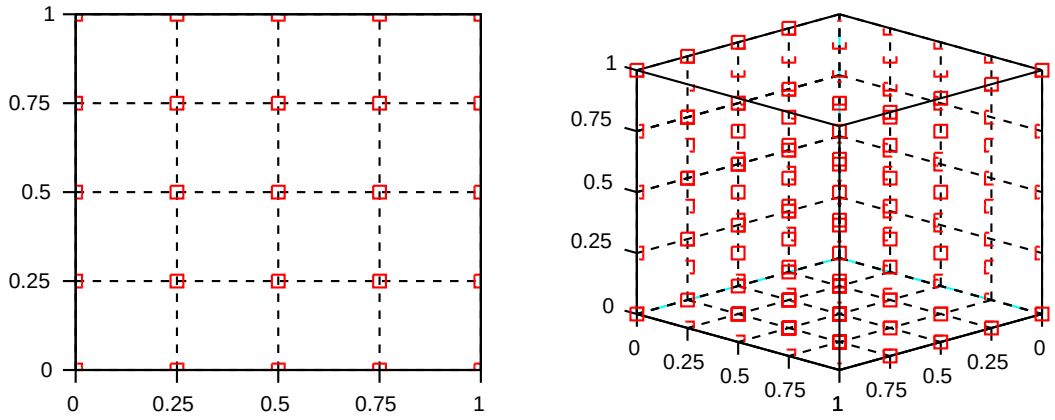
2.3 Nodes-based statistical model

As the behavior policies and the flocking task with target zone search have been already determined, the way to evaluate the swarm must be defined. In a multi-objective problem every solution can be represented as $\boldsymbol{x} \in \mathbb{R}^n$ and the evaluation of this solution can be represented as $\boldsymbol{f(x)} \in \mathbb{R}^m$, where n is the number of decision variables or the number of dimensions in the search space and m is the number of objective functions. When the multi-objective problem is represented by a simulator of a real-life process the evaluation of each candidate solution can be very expensive in terms of computational time, in addition if the process represented by the simulator contains stochastic factors the computational time problem is increased because each solution needs to be evaluated multiple times in order to obtain a more reliable value for every objective function. Multi-objective optimization techniques can be applied in this types of problems, but small number of population and generation or cycles must be used. In order to implement multi-objective optimizations techniques with this requirements in this type of problems a surrogate model that represents the simulator is needed.

2.3.1 Nodes generation

A very important part of this statistical model is the generation of the nodes, for that reason vectors which represents the positions of the nodes in each independent decision variable are determined as follows:

$$\begin{aligned}
\boldsymbol{p_1} &= [lb_1 \quad lb_1 + \Delta_1 \quad lb_1 + 2\Delta_1 \quad \dots \quad lb_1 + (N_1 - 1)\Delta_1] \\
\boldsymbol{p_2} &= [lb_2 \quad lb_2 + \Delta_2 \quad lb_2 + 2\Delta_2 \quad \dots \quad lb_2 + (N_2 - 1)\Delta_2] \\
&\vdots \\
\boldsymbol{p_n} &= [lb_n \quad lb_n + \Delta_n \quad lb_n + 2\Delta_n \quad \dots \quad lb_n + (N_n - 1)\Delta_n]
\end{aligned} \tag{2.29}$$



(a) Bi-dimensional search space

(b) Tri-dimensional search space

Figure 2.6: Examples of generated nodes

Where n is the number of dimensions in the search space, N_i is the desired number of nodes in the dimension i and Δ_i represents the space between nodes in the dimension i that can be calculated from the following equation:

$$\Delta_i = \frac{ub_i - lb_i}{N_i - 1} \tag{2.30}$$

Where ub_i and lb_i are the upper and lower bounds in the dimension i respectively. All the combinations between elements of different vectors of the equation (2.29) produces the position of the generated nodes in the search space. Every generated node is represented as a vector $\boldsymbol{v_k} = [v_{k,1}, v_{k,2}, \ldots, v_{k,n}]$ where the elements meet the condition $v_{k,i} \in \boldsymbol{p_i}$ and represent the position of the generated node k in the dimension i. The matrix $\boldsymbol{V} = [\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_K}]^T$ contains all the linear positions of the generated nodes. With this methodology K nodes uniformly distributed in all the search space are generated, where $K = \prod_{i=1}^{n} N_i$. In the Figure 2.6 two examples of generated nodes using $N_i = 5$, $lb_i = 0$, $ub_i = 1$ $\forall i$ are showed.

As can be seen in the Figure 2.6 the generated nodes generate n-orthotopes uniformly distributed in the search space.

2.3.2 NODES EVALUATION

This process is the most expensive part of the proposed model in terms of computational time. When the problem which will be optimized includes stochastic factors every generated node must be evaluated for multiple runs, the vector $\boldsymbol{f(v_k)} = [f_{1,k}, f_{2,k}, \ldots, f_{m,k}]$ contains the mean evaluation in each objective function obtained by the node k, where m is the number of objective functions and $f_{j,k}$ is the value of the j objective function obtained by the generated node k. The matrix $\boldsymbol{F_{nodes}} = [\boldsymbol{f(v_1)}, \boldsymbol{f(v_2)}, \ldots, \boldsymbol{f(v_K)}]^T$ contains the evaluation of every generated node. As shown in the Figure 2.6a in the bi-dimensional case 25 generated nodes are evaluated and as shown in the Figure 2.6b in the tri-dimensional case 125 generated nodes are evaluated. In this way only the nodes are evaluated for multiple runs using the quadrotor swarm simulator shown in the Algorithm 2.1. With this information available any solution inside the search space can be estimated from the known evaluation of the generated nodes. The process to estimate the evaluation of any solution from the nodes evaluation is described in the following section.

2.3.3 ESTIMATED EVALUATION

Suppose a vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ that represents a solution inside the search space with unknown evaluation of its objective functions. The matrix which represents the positions of the two neighbors for the solution $\boldsymbol{x}$ in each dimension is represented by:

$$\boldsymbol{O} = \begin{bmatrix} \max\limits_{1 \leq j \leq N_1} \{p_{1,j} \mid p_{1,j} \leq x_1\} & \min\limits_{1 \leq j \leq N_1} \{p_{1,j} \mid p_{1,j} > x_1\} \\ \max\limits_{1 \leq j \leq N_2} \{p_{2,j} \mid p_{2,j} \leq x_2\} & \min\limits_{1 \leq j \leq N_2} \{p_{2,j} \mid p_{2,j} > x_1\} \\ \vdots & \vdots \\ \max\limits_{1 \leq j \leq N_n} \{p_{n,j} \mid p_{n,j} \leq x_n\} & \min\limits_{1 \leq j \leq N_n} \{p_{n,j} \mid p_{n,j} > x_n\} \end{bmatrix} \tag{2.31}$$

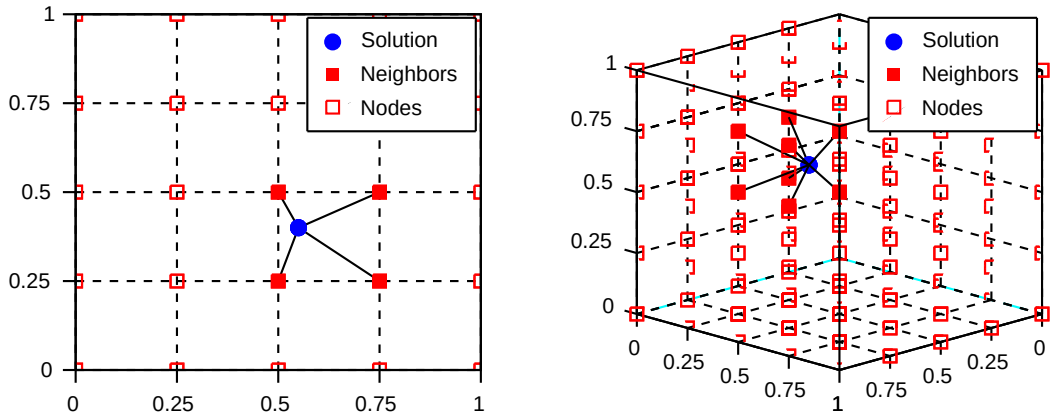
Where $p_{i,j}$ is the element j from the vector $\boldsymbol{p_i}$ of the equation (2.29). The rows of the matrix $\boldsymbol{O}$ can be determined as follows:

$$\boldsymbol{O} = \begin{bmatrix} \boldsymbol{o_1} \\ \boldsymbol{o_2} \\ \vdots \\ \boldsymbol{o_n} \end{bmatrix} = \begin{bmatrix} o_{1,1} & o_{1,2} \\ o_{2,1} & o_{2,2} \\ \vdots & \vdots \\ o_{n,1} & o_{n,2} \end{bmatrix} \tag{2.32}$$

Where $\boldsymbol{o}_i$ is the vector that contains the location of the two neighbors in the dimension i and $o_{i,j}$ represents the location the neighbor j in the dimension i. By combining elements of different rows of $\boldsymbol{O}$ it is possible to produce 2^n combinations that represent the linear position in the search space of the neighbors nodes. Each neighbor node is represented as a vector $\boldsymbol{u_s} = [u_{s,1}, u_{s,2}, \ldots, u_{s,n}]$, where the elements of $\boldsymbol{u_s}$ meet the condition $u_{s,i} \in \boldsymbol{o_i}$ and represents the position of the neighbor node s in the dimension i. The matrix $\boldsymbol{U} = [\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_{2^n}}]^T$ is determined in order to contain all the linear positions of the neighbors nodes. The evaluations of the neighbors nodes are already contained in the matrix $\boldsymbol{F_{nodes}}$ as every selected neighbor node $\boldsymbol{u_s}$ is equal to an already generated node $\boldsymbol{v_k}$, therefore the objective functions of each selected neighbor node can be extracted from the objective functions of the generated nodes as represented by the following equation:

$$\boldsymbol{f(u_s)} = \boldsymbol{f(v_k)} \Leftrightarrow \boldsymbol{u_s} = \boldsymbol{v_k} \tag{2.33}$$

The matrix $\boldsymbol{F_{neighbors}} = [\boldsymbol{f(u_1)}, \boldsymbol{f(u_2)}, \ldots, \boldsymbol{f(u_{2^n})}]^T$ contains only the objective functions of the neighbors nodes. In the Figure 2.7 can be seen two examples of the selection of neighbors nodes from a solution $\boldsymbol{x}$ using $N_i = 5$, $lb_i = 0$, $ub_i = 1$ $\forall i$. As shown in the Figure 2.7a in the bi-dimensional case 4 neighbors nodes are selected and as shown in the Figure 2.7b in the tri-dimensional case 8 neighbors nodes are selected.



(a) Bi-dimensional search space

(b) Tri-dimensional search space

Figure 2.7: Examples of selected neighbors nodes

As can be seen in the Figure 2.7 it is clearly that any solution in the search space is surrounded by 2^n neighbors nodes, this condition meets in search spaces with higher decision variables, the proximity of the the solution with unknown evaluation to each neighbor node is very important in order to estimate its objective functions. The euclidean distances between the solution $\boldsymbol{x}$ and the 2^n selected neighbors nodes are calculated as follows:

$$d_s = \sqrt{\sum_{i=1}^{n} (x_i - u_{s,i})^2} \tag{2.34}$$

The vector $\boldsymbol{d} = [d_1, d_2, \ldots, d_{2^n}]$ contains the euclidean distances between the solution $\boldsymbol{x}$ and the neighbors nodes. Using the vector $\boldsymbol{d}$ is generated a vector of weights $\boldsymbol{w} = [w_1, w_2, \ldots, w_{2^n}]$, each element of the vector $\boldsymbol{w}$ is calculated from euclidean distances by using the "SoftMax" function described by Sutton et al. [57] with a negative exponent as follows:

$$w_s = \frac{e^{-d_s/\tau}}{\sum_{k=1}^{2^n} e^{-d_k/\tau}} \tag{2.35}$$

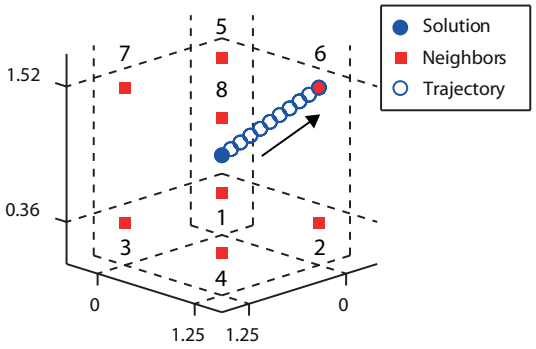


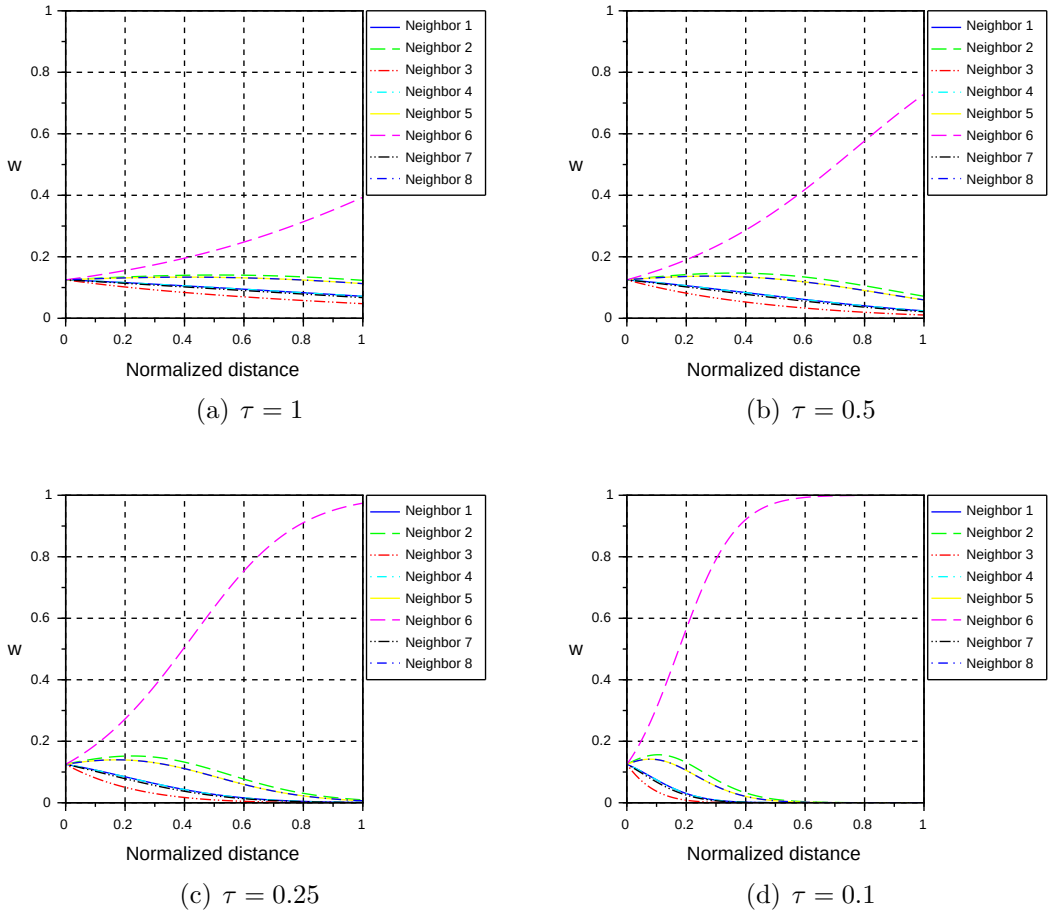
Figure 2.8: Solution in different positions

Where w_s is the calculated weight for the neighbor node $\boldsymbol{u_s}$ and τ is a positive parameter called the temperature which determines the influence of the euclidean distances in the weights generation. This transformation of euclidean distances into weights has the following properties.

- $\sum_{s=1}^{2^n} w_s = 1$.
- In the limit as $\tau \to 0$ the smallest distance produces $w \approx 1$.
- In the limit as $\tau \to \infty$ all the weights are equal regardless the distance.

The weights distribution of neighbors nodes can be adjusted by modifying the τ parameter. Figure 2.8 shows a case for a solution in the center of its neighbors and moves toward a particular neighbor, this example is useful to verify the evolution of the calculated weights of neighbors and its distribution trough this trajectory by using the "SoftMax" function.

Figure 2.9 shows the weights evolution for each neighbor when a solution is moved through different positions showed in the Figure 2.8. The weights are calculated using different values of τ parameter, is clearly that large values of τ produces a more balanced distribution of weights regardless the proximity of neighbors and for small values of τ the proximity of neighbors is very important for the distribution of weights.



(a) $\tau = 1$ (b) $\tau = 0.5$

(c) $\tau = 0.25$ (d) $\tau = 0.1$

Figure 2.9: Weights calculated with different values of τ

The estimated objective functions of the solution $\boldsymbol{x}$ are calculated by a linear combination between the weights and the objective functions of the selected neighbors nodes, this produces estimated objective functions that depend on the proximity of the neighbors nodes to the solution to be estimated, as showed in the following equation:

$$\boldsymbol{f}(\boldsymbol{x})^T = \begin{bmatrix} f_{1,x} \\ f_{2,x} \\ \vdots \\ f_{m,x} \end{bmatrix} = \begin{bmatrix} w_1 f_{1,1} + w_2 f_{1,2} + \cdots + w_{2^n} f_{1,2^n} \\ w_1 f_{2,1} + w_2 f_{2,2} + \cdots + w_{2^n} f_{2,2^n} \\ \vdots \\ w_1 f_{m,1} + w_2 f_{m,2} + \cdots + w_{2^n} f_{m,2^n} \end{bmatrix} \tag{2.36}$$

Where $f_{j,x}$ is the estimated objective function j of the solution $\boldsymbol{x}$ and $f_{j,s}$ is the objective function j of the neighbor node $\boldsymbol{u_s}$. The equation (2.36) can be calculated with the following operation:

$$\boldsymbol{f(x)} = \boldsymbol{w} * \boldsymbol{F_{neighbors}} \tag{2.37}$$

With this linear combination, the objective functions of the solution $\boldsymbol{x}$ are estimated from the selected neighbors nodes, where closer neighbors has a bigger influence over the estimated evaluation of $\boldsymbol{x}$, and this influence of closer neighbors can be adjusted for each multi-objective problem using the parameter τ in order to obtain a better quality of estimation. The estimation performance is expected to be better when smooth objective functions are estimated.

2.3.4 ALGORITHM OF NODES-BASED STATISTICAL MODEL

The proposed statistical model is summarized in the Algorithm 2.2 as an alternative from the simulator represented in the Algorithm 2.1 to evaluate solutions, however in order to apply this statistical model it is necessary to evaluate the required nodes directly from the simulator.

Algorithm 2.2 Pseudocode of Nodes-based statistical model

Input: $\boldsymbol{x}$, $\boldsymbol{V}$, $\boldsymbol{F_{nodes}}$;
Output: $\boldsymbol{f(x)}$;

Detect two neighbors in each dimension as equation (2.31);
Obtain matrix $\boldsymbol{U}$ with linear positions of neighbors nodes;
Obtain matrix $\boldsymbol{F_{neighbors}}$ by using equation (2.33);
for $s = 1 : 2^n$ **do**
    Calculate distances between solution and neighbors as equation (2.34);
end for
for $s = 1 : 2^n$ **do**
    Transform distances into weights as equation (2.35);
end for
Estimate objective functions of solution $\boldsymbol{x}$ as equation (2.37);

2.4 Multi-objective optimization

This section is focused on explaining the concepts of multi-objective optimization and explain what are the implications of find a optimial solution of a problem with multiple objectives that must be satisfy at the same time. Then a review of three well known multi-objective optimization algorithms that are implemented in this thesis are given, the modifications in some algorithms are explained in order to enable the replication of the results.

2.4.1 Basic concepts

The optimal solution of a multi-objective problem is not closed to a single value, in this type of problems, the solutions can not be compared by relational operators as in single-objective optimization. In this case, to decide if a solution is better than another is necessary determine if one solution dominates the other, this means that all the objective values $\boldsymbol{f(x)}$ are better or equal than the objective values of other solution $\boldsymbol{f(y)}$, this is called Pareto dominance. The following definitions are formulated as a minimization problem:

Definition 1 *Suppose that* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ *are solutions in a search space of dimension* n *and* $\boldsymbol{f(x)}, \boldsymbol{f(y)} \in \mathbb{R}^m$ *contain the evaluation of* m *objective functions. The Pareto Dominance means that* $\boldsymbol{x}$ *dominates* $\boldsymbol{y}$ *(denoted as* $\boldsymbol{x} \succ \boldsymbol{y}$*) iff:*

$$\exists \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n \mid \forall i \ f_i(x) \leq f_i(y) \tag{2.38}$$

Definition 2 *The Pareto Optimality refers to a solution* $\boldsymbol{x_i^*} \in \mathbb{R}^n$ *that is called a nondominated solution iff:*

$$\nexists \boldsymbol{x_j} \in \mathbb{R}^n \mid \boldsymbol{x_j} \succ \boldsymbol{x_i^*} \ \forall j \tag{2.39}$$

Definition 3 *A set including all the nondominated solutions of a problem is called Solution Space (**SS**) and it is defined as the set that includes all the Pareto-optimal solutions, where n is the number of nondominated solutions found.*

$$\boldsymbol{SS} = \{\boldsymbol{x_1^*}, \boldsymbol{x_2^*}, ..., \boldsymbol{x_n^*}\} \tag{2.40}$$

Definition 4 *The Pareto front (**PF**) is the set which contains all the objective values of the nondominated solutions in the solution space.*

$$\boldsymbol{PF} = \{\boldsymbol{f}(\boldsymbol{x_i^*}) \mid \boldsymbol{x_i^*} \in \boldsymbol{SS}\} \tag{2.41}$$

The use of metaheuristics algorithms in hard optimization problems is a good alternative in comparison with classical optimization methods. However, optimal solutions are not guaranteed, despite this, very good approximations to the optimal are obtained with the use of metaheuristics optimization algorithms [58].

2.4.2 MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

The Multi-Objective Particle Swarm Optimization (MOPSO) is an extension of the Particle Swarm Optimization (PSO) that was proposed by Coello et al. [59] .

2.4.2.1 DESCRIPTION OF MOPSO

The position of the particles in the seach space are generated randomly and the initial velocity of the particles is selected as zero. The velocity in the following cycles is computed as formulated by Reyes-Sierra and Coello [60]:

$$\boldsymbol{v_i(t)} = w\boldsymbol{v_i(t-1)} + \beta_1 u_1(\boldsymbol{x_{b,i}} - \boldsymbol{x_i(t)}) + \beta_2 u_2(\boldsymbol{x_l} - \boldsymbol{x_i(t)}) \tag{2.42}$$

Where w is the inertial weight, β_1 is the personal learning coefficient, β_2 is the global learning coefficient, $u_1, u_2 \in [0, 1]$ are random values from a uniform distribution, $\boldsymbol{x_{b,i}}$ is the best found position of the particle i and $\boldsymbol{x_l}$ is the position of the leader taken from the repository, the leader is randomly selected from a hypercube generated by a previous designed grid in the search space which was selected by Roulette Wheel Selection, hypercubes with less particles has a higher probability to be selected. The inertial weight decreases linearly through the cycles by the damping rate parameter "s" as proposed by Shi and Eberhart [61]. The new position of the particle i is obtained using the following equation:

$$\boldsymbol{x_i(t) = x_i(t-1) + v_i(t)} \tag{2.43}$$

2.4.2.2 External repository

The external repository has the objective of store the nondominated particles along the cycles. The repository controller has the function of decide which particles maintain in the repository, maintaining only nondominated particles. The adaptive grid depends of the solutions in the repository and is used to well distribute the nondominated particles in the search space by selecting the leader from hypercubes with less particles and is useful to decide which particle delete when the repository is full from a hypercube which was selected by Roulette Wheel Selection, hypercubes with more particles has a higher probability to be selected.

2.4.2.3 Mutation operator

A mutation operator is necessary in the MOPSO due to the very high convergence speed of the PSO. The pseudocode of the implemented mutation operator [60] is given below:

Algorithm 2.3 Mutation operator in MOPSO

$n \leftarrow$ Dimension of the individual $\boldsymbol{x}$;
$i \leftarrow$ A random integer between 1 and n;
$lb \leftarrow$ Lower bound of dimension i;
$ub \leftarrow$ Upper bound of dimension i;
$m_{range} = pm \cdot (ub - lb)$;
$a = x[i] - m_{range}$;
$b = x[i] + m_{range}$;
if $a < lb$ **then**
    $a = lb$;
end if
if $b > ub$ **then**
    $b = ub$;
end if
$x[i] \leftarrow$ A random number between a and b;

This mutation operator minimize its occurrence and its effect as the cycles increases, the mutation probability pm is calculated as follows:

$$pm = \left(1 - \frac{Current\ cycle - 1}{Total\ cycles - 1}\right)^{\frac{5}{\mu}} \tag{2.44}$$

Where μ is the mutation rate. Figure 2.10 represents the effect of the mutation in the population as the cycles increases using $\mu = 0.5$.

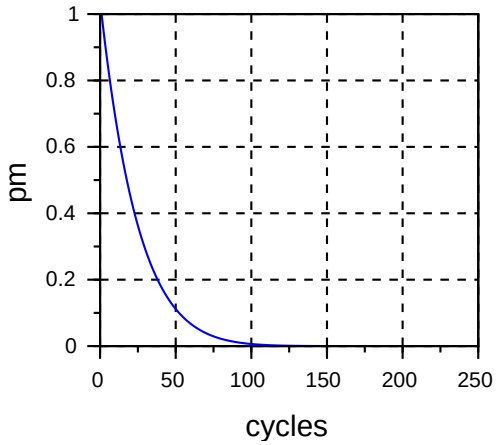


Figure 2.10: Evolution of the mutation probability

2.4.2.4 Pseudocode of MOPSO

The pseudocode of this algorithm is presented below:

Algorithm 2.4 Pseudocode of MOPSO

```
Initialize the population as random vectors in the search space;
Initialize the velocity of each particle as zero;
Determine best position of particles as current position;
Evaluate the position of every particle;
Verify non-dominated particles and update repository;
for t = 1 : Total cycles do
    for i = 1 : Total of particles do
        Select the position of the leader x_l from the repository;
        Calculate velocity of the particle v_i(t) as equation (2.42);
        Calculate position of the particle x_i(t) as equation (2.43);
        if Position of the particle is out of boundaries then
            Decision variables out takes the value of the boundary;
            v_i(t) = −v_i(t);
        end if
        Calculate mutation probability pm as equation (2.44);
        if A random number between 0 and 1 < pm then
            Apply mutation operator on position of particle;
        end if
        Evaluate new position x_i(t) of particle;
        if x_i(t) ≻ x_{b,i} then
            x_{b,i} = x_i(t);
        else if x_{b,i} ≻ x_i(t) then
            x_{b,i} = x_{b,i};
        else
            if A random number between 0 and 1 < 0.5 then
                x_{b,i} = x_i(t);
            end if
        end if
    end for
    Verify nondominated particles and update repository;
    if Solutions in repository > Size of repository then
        Delete particles as explained in Section 2.4.2.2;
    end if
    w = w − s;
end for
```

2.4.3 Nondominated Sorting Genetic Algorithm II with Differential Evolution

The Nondominated Sorting Genetic Algorithm II (NSGA-II) was presented by Deb et al. [62]. In this work the algorithm NSGA-II with Differential Evolution (NSGA-II-DE) proposed by Li and Zhang [63] is implemented and a repository is added for the nondominated solutions found during the optimization process.

2.4.3.1 Description of NSGA-II-DE

The initial population is randomly generated in the search space from a uniform distribution. The function FastNonDominatedSort($\boldsymbol{P}$) sorts the population by different nondomination levels; every solution is compared against all the other solutions, if a solution is a nondominated solution as describen in the equation (2.39) then a nondomination rank of one is assigned; if a solution is only dominated by solutions with nondomination rank of one then a nondomination rank of two is assigned and so on with all the solutions. The set of solutions with nondomination rank of one form the first Pareto front in the objective space, the set of solutions with nondomination rank of two form the second Pareto front and so on. The function CrowdingDistanceAssignment($\boldsymbol{r}$) assigns a crowding value to each individual based on the proximity of other individuals from the same nondomination level, the crowding value assigned to a solution is obtained by calculate the distance between its two closest neighbors with the same nondomination rank in each objective, a small crowding value represents that the solution is more crowded by other solutions. The function CrowdedComparisonOperator($\boldsymbol{r}, \boldsymbol{d}$) is called in order to sort all the population in descending order using the operator $\succ_n$ which is described by the following equation:

$$i \succ_n j \text{ if } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_d > j_d)) \tag{2.45}$$

Where i_{rank}, j_{rank} are the nondomination rank and i_d, j_d are the crowding distance of the individuals i and j respectively. The population of the following generation is generated using differential evolution where the first individuals of the sorted population $\boldsymbol{P}$ are better evaluated.

2.4.3.2 Differential Evolution

The pseudocode for the differential evolution is described as follows:

Algorithm 2.5 Differential evolution

```
P ← Initial population;
CR ← Crossover probability;
F ← Scaling factor;
for i = 1 : Size of population do
    [v1, v2, v3] ← Individuals selected from P;
    for j = 1 : Dimension of the individuals do
        if A random number between 0 and 1 < CR then
            Poi[j] = v1[j] + F (v2[j] − v3[j]);
        else
            Poi[j] = v1[j];
        end if
    end for
    Poi =PolynomialMutation(Poi);
    if Individual is out of boundaries then
        Decision variables out of boundaries takes a random value;
    end if
end for
```

Where $\boldsymbol{Po}$ is the offspring population with the same size as $\boldsymbol{P}$ and the selection of $\boldsymbol{v_1}$, $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$ is done using a binary tournament selection over the sorted population $\boldsymbol{P}$, where the first members are the best evaluated and have a bigger possibility to be selected.

2.4.3.3 Polynomial mutation operator

The pseudocode of the polynomial mutation operator is given below:

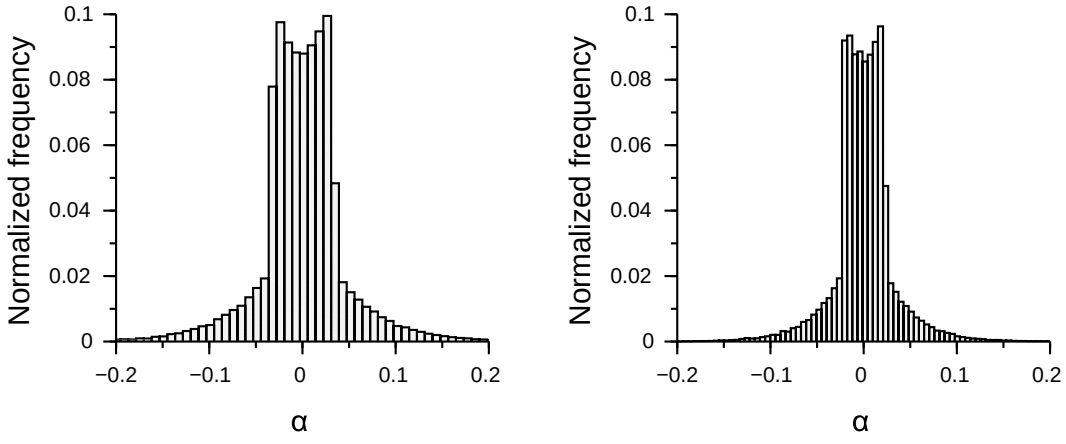
Algorithm 2.6 Polynomial mutation operator

```
pm ← Mutation probability;
for i = 1 : Dimension of the individuals do
    if A random number between 0 and 1 < pm then
        lb ← Lower bound of dimension i;
        ub ← Upper bound of dimension i;
        Calculate α as equation (2.46);
        x[i] = x[i] + α(ub − lb);
    end if
end for
```

Where α is calculated every time that an individual is selected for mutation using the following expression:

$$\alpha = \begin{cases} (2u)^{\frac{1}{\eta+1}} - 1, & \text{A random number} < 0.5 \\ 1 - (2-2u)^{\frac{1}{\eta+1}}, & \text{Otherwise} \end{cases} \tag{2.46}$$

Where $u \in [0, 1]$ is a random number and η is the distribution index. In the Figure 2.11 the histogram for 100000 generated values of α using different values for η is showed, it is clear that smaller values for η produces aggresive mutations.



(a) $\eta = 20$ (b) $\eta = 30$

Figure 2.11: Distribution of α

2.4.3.4 Pseudocode of NSGA-II-DE

The pseudocode is presented below, where $\boldsymbol{r}$ and $\boldsymbol{d}$ are vectors with ranks and crowding distances respectively and $\boldsymbol{rank_1}$ contains the solutions in the first front.

Algorithm 2.7 Pseudocode of NSGA-II-DE

$N \leftarrow$ Size of population;
$N_{Rep} \leftarrow$ Maximum size of repository;
Initialize a random population $\boldsymbol{P}$ in the search space;
Evaluate each individual in the population $\boldsymbol{P}$;
$\boldsymbol{r}$ =FastNonDominatedSort($\boldsymbol{P}$);
$\boldsymbol{d}$ =CrowdingDistanceAssignment($\boldsymbol{r}$);
$\boldsymbol{P}$ =CrowdedComparisonOperator($\boldsymbol{r}, \boldsymbol{d}$);
$\boldsymbol{P_{Rep}} = \boldsymbol{P} \in r_1$;
for $t = 1$: Total generations **do**
    $\boldsymbol{Q}$ =DifferentialEvolution($\boldsymbol{P}$);
    Evaluate each individual in the population $\boldsymbol{Q}$;
    $\boldsymbol{R} = \boldsymbol{P} \cup \boldsymbol{Q}$;
    $\boldsymbol{r}$ =FastNonDominatedSort($\boldsymbol{R}$);
    $\boldsymbol{d}$ =CrowdingDistanceAssignment($\boldsymbol{r}$);
    $\boldsymbol{R}$ =CrowdedComparisonOperator($\boldsymbol{r}, \boldsymbol{d}$);
    $\boldsymbol{P} = \boldsymbol{R}[\boldsymbol{1} : \boldsymbol{N}]$;
    $\boldsymbol{P_{Rep}} = \boldsymbol{P_{Rep}} \cup \boldsymbol{P} \in \boldsymbol{rank_1}$;
    $\boldsymbol{r}$ =FastNonDominatedSort($\boldsymbol{P_{Rep}}$);
    $\boldsymbol{d}$ =CrowdingDistanceAssignment($\boldsymbol{r}$);
    $\boldsymbol{P_{Rep}}$ =CrowdedComparisonOperator($\boldsymbol{r}, \boldsymbol{d}$);
    $\boldsymbol{P_{Rep}} = \boldsymbol{P_{Rep}} \in \boldsymbol{rank_1}$;
    if Individuals in $\boldsymbol{P_{Rep}} > N_{Rep}$ **then**
        $\boldsymbol{P_{Rep}} = \boldsymbol{P_{Rep}}[\boldsymbol{1} : \boldsymbol{N_{Rep}}]$;
    end if
end for

2.4.4 Multiobjective Evolutionary Algorithm based on Decomposition with Differential Evolution

Zhang and Li [64] proposed the Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) and then a new version of this algorithm using differential evolution (MOEA/D-DE) was proposed [63]. This last algorithm is implemented in this study with the addition of the repository for preserve the nondominated so-

lutions found through the generations and the adaptive grid from MOPSO which deletes solutions when the repository is full while maintains the nondominated solutions well distributed along the Pareto front.

2.4.4.1 DESCRIPTION OF MOEAD/D-DE

Weight vectors are generated from an integer H as combinations of elements of the following set.

$$\left\{ \frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H} \right\} \tag{2.47}$$

Every weight vector is represented as $\boldsymbol{\lambda_i} = [\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,m}]$, where m is the number of objectives, $\lambda_{i,j} \geq 0 \ \forall i, j$ and $\sum_{j=1}^{m} \lambda_{i,j} = 1$. The number of subproblems N are calculated as equation (2.48) and is determined by the selection of the integer H and the number of objectives m.

$$N = C_{H+m-1}^{m-1} \tag{2.48}$$

The algorithm which generates weight vectors in a multi-objective problem with 4 objectives can be done with nested for loops as presented below:

Algorithm 2.8 Algorithm to generate weight vectors with $m = 4$

```
v = [0 : 1/H : 1];
s = 1;
for i = 1 : H + 1 do
    for j = 1 : H + 2 − i do
        for k = 1 : H + 3 − i − j do
            λ_s = [v_i, v_j, v_k, v_{H+4−i−j−k}];
            s = s + 1;
        end for
    end for
end for
```

A total of N solutions $\boldsymbol{x_i} = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]$ are generated randomly inside the search space, where n is the number of decision variables. The reference point $\boldsymbol{z} = [z_1, z_2, \ldots, z_m]$ is a vector with the best value for every objective function. Supposing a minimization problem, every element of $\boldsymbol{z}$ is calculated as follows.

$$z_j = \min_{1 \leq i \leq N} \{f_j(\boldsymbol{x_i})\} \tag{2.49}$$

Where $f_j(\boldsymbol{x_i})$ is the objective function j of the solution $\boldsymbol{x_i}$. The elements of $\boldsymbol{z}$ are updated by comparing its current values with the evaluations of the new proposed solutions $\boldsymbol{y}$.

$$z_j = \min\{z_j, f_j(\boldsymbol{y})\} \tag{2.50}$$

The main idea of the MOEA/D-DE is to decompose a multi-objective problem into scalar subproblems and optimize them simultaneosly. The Tchebycheff aproach is implemented for this decomposition as follows:

$$g(\boldsymbol{x_i} \mid \boldsymbol{\lambda_i}, \boldsymbol{z}) = \max_{1 \leq j \leq m} \{\lambda_{i,j} \mid f_j(\boldsymbol{x_i}) - z_j \mid\} \tag{2.51}$$

2.4.4.2 DIFFERENTIAL EVOLUTION AND EXTERNAL REPOSITORY

The pseudocode for the differential evolution is the same as the Algorithm 2.5 but the offspring is generated mainly between solutions with closest weight vectors and only one solution is obtained per function call. The method which controls the repository with adaptive grid is the same as explained in the Section 2.4.2.2.

2.4.4.3 Pseudocode of MOEAD/D-DE

The pseudocode of this algorithm is presented below, where T is the total of neighbors, δ is the probability to select parents from neighborhood and nr is the maximum number of solutions replaced by a child.

Algorithm 2.9 Pseudocode of MOEA/D-DE

Generate N weight vectors $\boldsymbol{\lambda_i}$;
Generate N random solutions $\boldsymbol{x_i}$ for every weight vector;
Calculate euclidean distances between weight vectors;
for $i = 1 : N$ **do**
    Evaluate solution $\boldsymbol{x_i}$;
    Select T closest weight vectors from $\boldsymbol{\lambda_i}$;
    $\boldsymbol{b_i} \leftarrow T$ Solutions indexes with closest weight vectors from $\boldsymbol{\lambda_i}$;
end for
Update reference point $\boldsymbol{z}$ as the equation (2.49);
Verify nondominated solutions and update repository;
for $t = 1 :$ Total generations **do**
    for $i = 1 : N$ **do**
        $\boldsymbol{v_1} = \boldsymbol{x_i}$;
        if A random number between 0 and 1 $< \delta$ **then**
            Select $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$ randomly from neighborhood;
            $\boldsymbol{d} = \boldsymbol{b_i}$;
        else
            Select $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$ randomly from all solutions;
            $\boldsymbol{d} = \{1, 2, ..., N\}$;
        end if
        $\boldsymbol{y} =$DifferentialEvolution$([\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}])$;
        Evaluate the new solution $\boldsymbol{y}$;
        Update reference point $\boldsymbol{z}$ as the equation (2.50);
        for $j = 1 : nr$ **do**
            Select a random index k from $\boldsymbol{d}$;
            if $g(\boldsymbol{y} \mid \boldsymbol{\lambda_k}, \boldsymbol{z}) \leq g(\boldsymbol{x_k} \mid \boldsymbol{\lambda_k}, \boldsymbol{z})$ **then**
                $\boldsymbol{x_k} = \boldsymbol{y}$;
                $f(\boldsymbol{x_k}) = f(\boldsymbol{y})$;
            end if
        end for
    end for
    Verify nondominated solutions and update repository;
end for

2.5 Comparison techniques

This section presents the techniques that are implemented in order to make the necessary comparisons about the obtained results. First, the selected metrics to evaluate the performance of the multi-objective optimization algorithms are explained, then the bootstrap methods that are used to verify the correct calculation and estimation of objective functions during the optimization process are presented, finally the multivariate normal distribution is explained in order to compare the relations between control parameters and objective functions.

2.5.1 Performance metrics for multi-objective optimization algorithms

Various techniques are proposed in order to compare the performance multi-objective optimization algorithms [65]. The following metrics are selected in this work for this purpose.

2.5.1.1 Hypervolume measure

Given a solution space ($\boldsymbol{SS}$) and a reference point $\boldsymbol{z_{ref}} = [z_{ref,1}, z_{ref,2}, \ldots, z_{ref,m}]$ which is dominated by all the points in the Pareto front $\boldsymbol{PF} = \{\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_{|PF|}}\}$ generated by the $\boldsymbol{SS}$ is possible to calculate the region between $\boldsymbol{PF}$ and $\boldsymbol{z_{ref}}$, this is originally named "size of the space covered" [66] and then Hypervolume measure [67]. In the Figure 2.12 an example of this region in a bi-dimensional case considering a minimization problem is showed, where the point $\boldsymbol{z_{ref}}$ can be given by any point which is dominated by all the points in the Pareto front, in this thesis the point $\boldsymbol{z_{ref}}$ is selected by using two different methodologies [68, 69].

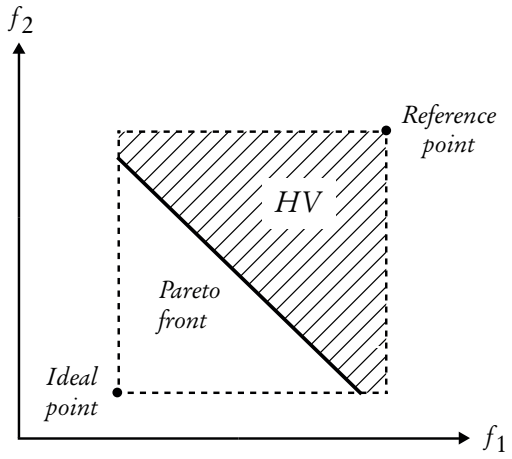


Figure 2.12: Hypervolume measure in a bi-dimensional case

The methodology explained by Knowles [70] to calculate the Hypervolume measure is used in this thesis. Given a point in the Pareto front $\boldsymbol{z_i} = [z_{i,1}, z_{i,2}, \ldots, z_{i,m}]$ and a reference point $\boldsymbol{z_{ref}}$, the region generated between the points $\boldsymbol{z_i}$ and $\boldsymbol{z_{ref}}$ is given by the following set:

$$HV(\boldsymbol{z_i}, \boldsymbol{z_{ref}}) = \{\boldsymbol{y} \mid \boldsymbol{y} < \boldsymbol{z_{ref}} \; and \; \boldsymbol{z_i} < \boldsymbol{y}, \; \boldsymbol{y} \in \mathbb{R}^m\} \tag{2.52}$$

The generated Hypervolume measure obtained by $\boldsymbol{PF}$ and $\boldsymbol{z_{ref}}$ is given by the Lebesgue integral of the following set:

$$HV(\boldsymbol{PF}, \boldsymbol{z_{ref}}) = \bigcup_{i=1}^{|\boldsymbol{PF}|} HV(\boldsymbol{z_i}, \boldsymbol{z_{ref}}) \tag{2.53}$$

The following expression is used to calculate the hypervolume in a problem with two objectives.

$$HV(\boldsymbol{PF}, \boldsymbol{z_{ref}}) = \sum_{i=1}^{|\boldsymbol{PF}|} (\mid z_{i,1} - z_{ref,1} \mid \cdot \mid z_{i,2} - z_{i-1,2} \mid) \tag{2.54}$$

Where $z_{0,2}$ is initially set as $z_{ref,2}$ and the points in the Pareto fronts are sorted in decreasing order in the objective 1. The function HYPERVOLUMEMEASURE($\boldsymbol{PF}, \boldsymbol{z_{ref}}, m$) can be used to calculate the Hypervolume measure in problems with m objectives as showed in the following pseudocode:

Algorithm 2.10 Hypervolume measure

```
1: HV = 0;
2: z_prev = z_ref;
3: while PF ≠ ∅ do
4:     PF ← Nondominated vectors with respect of objective m from PF;
5:     z_high ← Vector with largest value in objective m from PF;
6:     if m < 3 then
7:         HV_aux =| z_{high,m-1} − z_{prev,m-1} |;
8:     else
9:         HV_aux =HypervolumeMeasure(PF, z_ref, m − 1);
10:    end if
11:    HV = HV + (HV_aux · | z_{high,m} − z_{prev,m} |);
12:    z_{prev,m} = z_{high,m};
13:    PF = PF \ {z_r | z_{r,m} ≥ z_{high,m}, z_r ∈ PF};
14: end while
```

In most cases is desirable a normalized value for the Hypervolume measure, this can be done using the following equation:

$$HV_{normalized} = \frac{HV}{HV_{ideal}} \tag{2.55}$$

Where a point $\boldsymbol{z_{ideal}}$ is needed, in this work the point $\boldsymbol{z_{ideal}}$ is given by the best found value in each objective. The value for HV_{ideal} can be calculated using the following equation:

$$HV_{ideal} = \prod_{i=1}^{m} \mid z_{ref,i} - z_{ideal,i} \mid \tag{2.56}$$

2.5.1.2 Modified C-metric

The following metric which is based on the C-metric, originally named "coverage of two sets" [66], is proposed in order to evaluate the performance of the multi-objective optimization algorithms when the true Pareto front is unknown. Suppose that n multi-objective optimization algorithms are compared, this means that n solution spaces with the nondominated solutions found by every algorithm are available. The following equation shows the combined set ($\boldsymbol{CS}$) that includes the solution space of every algorithm.

$$\boldsymbol{CS} = \bigcup_{i=1}^{n} \boldsymbol{SS_i} \tag{2.57}$$

Where $\boldsymbol{SS_i}$ is the solution space found by the algorithm i. Every individual solution space includes nondominated solutions but nondominated solutions of an algorithm may dominate nondominated solutions from other algorithm. The solutions of $\boldsymbol{CS}$ denoted as $\boldsymbol{c_k}$ and the nondominated solutions of $\boldsymbol{CS}$ denoted as $\boldsymbol{c_j^*}$ follows the following equation:

$$\nexists \boldsymbol{c_k} \neq \boldsymbol{c_j^*} \in \boldsymbol{CS} \mid \boldsymbol{c_k} \succ \boldsymbol{c_j^*} \; \forall k \tag{2.58}$$

The set which includes only nondominated solutions $\boldsymbol{c_j^*}$ from $\boldsymbol{CS}$ is called combined solution space ($\boldsymbol{CSS}$) as can be seen as follows:

$$\boldsymbol{CSS} = \{\boldsymbol{c_1^*}, \boldsymbol{c_2^*}, ..., \boldsymbol{c_m^*}\} \tag{2.59}$$

Where m is the total of nondominated solutions in $\boldsymbol{CSS}$. The relationship between the number of solutions contained in $\boldsymbol{SS_i} \in \boldsymbol{CSS}$ and the number of solutions in $\boldsymbol{SS_i}$ is called quality of solutions of the algorithm i denoted as Q_i.

$$Q_i = \frac{| \boldsymbol{SS_i} \in \boldsymbol{CSS} |}{| \boldsymbol{SS_i} |} \tag{2.60}$$

The values of Q closer to 1 are preferable.

2.5.1.3 Computational time

The computational time per cycle denoted as TPC is also obtained as another performance metric. This metric shows the required time in seconds for an algorithm to complete a cycle in the optimization process. All the optimizations were running in 64-bits Windows 7 Ultimate, with a processor Intel Core i7-6700 CPU @ 3.40GHz and 16 GB of RAM.

2.5.2 Bootstrap methods

If the optimization process is done using the quadrotor swarm simulator explained in the Section 2.2.3 as evaluation function, a reduce number of runs per solution are proposed mainly by the computational time required per evaluation, for that reason a study based on bootstrap methods [71] is proposed to secure that the selected number of runs represent in a correct way the real evaluation of the objective functions. If the optimization process is done using the nodes-based statistical model explained in the Section 2.3.4 as evaluation function, the same study based on bootstrap methods can be done in order to verify the performance of the estimated evaluations. In our experiments, different nondominated solutions per experiment are selected in order to calculate its bootstrap t confidence intervals, if the obtained objective functions are inside of the calculated bootstrap t confidence intervals means that the obtained objective functions during the optimization process represent a tendency of the real evaluation of the solutions.

The bootstrap methods are based on B resamples of an original sample, the selected nondominated solution is evaluated for a total of R runs, this is the original sample which consists on four original samples each one corresponding of an objective function, the following procedure must be done for every objective function. The mean is denoted by $\overline{x}$, the mean of every resample is denoted by $\overline{x}_i^*$. The bootstrap mean and the standard error are calculated as follows:

$$mean_{boot} = \frac{1}{B}\sum_{i=1}^{B}\overline{x}_i^* \tag{2.61}$$

$$SE_{boot} = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(\overline{x}_i^* - mean_{boot}\right)^2} \tag{2.62}$$

Another important value is the *bias* which is the difference between the mean of the original sample and the bootstrap mean.

$$bias = \overline{x} - mean_{boot} \tag{2.63}$$

When the bootstrap distribution, which is the mean distribution of resamples, is approximately normal and the bias value is small, a bootstrap t confidence interval of the mean statistic can be calculated as denoted in the following equation:

$$\overline{x} \pm t \cdot SE_{boot} \tag{2.64}$$

Where t is the critical value of the t(n-1) distribution, and can be obtained from a t-table for different percentages of confidence intervals, in this study a 95% confidence interval is calculated. The requirement of the bootstrap distribution with normal shape is well satisfied by the objective functions f_1 and f_2, but the objective functions f_3 and f_4 in some cases has difficulties in satisfy this requirement, however

this is due to the consistency of the evaluations of these objective functions and this harms the correct shape of the bootstrap distribution. Nevertheless, this study is done for all the four objective functions supposing that the two requirements to calculate the bootstrap t confidence interval are satisfied.

In this thesis the selected size of the original sample is $R = 30$, and the total of resamples is selected as $B = 10000$.

2.5.3 MULTIVARIATE NORMAL DISTRIBUTION

In order to observe the relationship between control parameters and objective functions the generation of multivariate normal random values as described by Gentle [72] is proposed. The multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is described by a mean vector $\boldsymbol{\mu}$ of size n and a covariance matrix $\boldsymbol{\Sigma}$ of size $[n \times n]$ which are calculated from a set of observations $\boldsymbol{X}$ of size $[o \times n]$, where o is the number of observations and n is the number of variates. When the covariance matrix $\boldsymbol{\Sigma}$ is positive semi-definite can be decomposed as follows:

$$\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T \tag{2.65}$$

The matrix $\boldsymbol{A}$ can be calculated from $\boldsymbol{\Sigma}$ by LDL^T decomposition, where the covariance matrix $\boldsymbol{\Sigma}$ is decomposed in an unit lower triangular matrix $\boldsymbol{L}$ and a diagonal matrix $\boldsymbol{D}$.

$$\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T \tag{2.66}$$

The elements of the matrix $\boldsymbol{L}$ and $\boldsymbol{D}$ can be computed as the equations (2.67) and (2.68) respectively as mentioned by Krishnamoorthy and Menon [73].

$$L_{i,j} = \frac{1}{D_j}\left(\Sigma_{i,j} - \sum_{k=1}^{j-1} L_{i,k}L_{j,k}D_k\right) for\ i > j \tag{2.67}$$

$$D_i = \Sigma_{i,i} - \sum_{k=1}^{i-1} L_{i,k}^2 D_k \tag{2.68}$$

Therefore, a suitable matrix $\boldsymbol{A}$ can be calculated as the following equation:

$$\boldsymbol{A} = \boldsymbol{L}\sqrt{\boldsymbol{D}} \tag{2.69}$$

The equation (2.70) shows a vector $\boldsymbol{x}$ of size n with multivariate normal random values with distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{z} \tag{2.70}$$

Where $\boldsymbol{z}$ is a vector of size n with random values taken from the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma^2 = 1$, the elements of the vector $\boldsymbol{z}$ can be computed as follows:

$$z_i = \left(\sum_{j=1}^{12} u_j\right) - 6 \tag{2.71}$$

Where $u_j \in [0, 1]$ is a random number from an uniform distribution, the elements of the vector $\boldsymbol{x}$ can be standardized using the following equation:

$$\boldsymbol{x_s} = (\boldsymbol{x} - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma} \tag{2.72}$$

Where $\oslash$ represents an element-wise division operator and $\boldsymbol{\sigma}$ is a standard deviations vector of size n taken from a set of observations $\boldsymbol{X}$.

Chapter 3

Experimental Setup

3.1 Proposed experiments

The different combinations between swarm sizes and conditions of the arena to be optimized are proposed in this section, all the proposed combinations are optimized using two different approaches; the first set of experiments are done using the quadrotor swarm simulator for obtain the objective functions of the candidate solutions found during the optimization process, the second set of experiments are done over the nodes-based statistical model for estimate the objective functions of the candidate solutions found during the optimization process.

3.1.1 Using quadrotor swarm simulator

In this set of experiments the multi-objective optimization algorithms evaluate the solutions directly using the quadrotor swarm siulator (QSS) from the Section 2.2.3, this process has a high computational cost because each candidate solution is evaluated 10 repetitions in order to have a good approximation of its objective functions, for that reason only small populations and a reduce number of cycles are suitable. The following experiments are proposed:

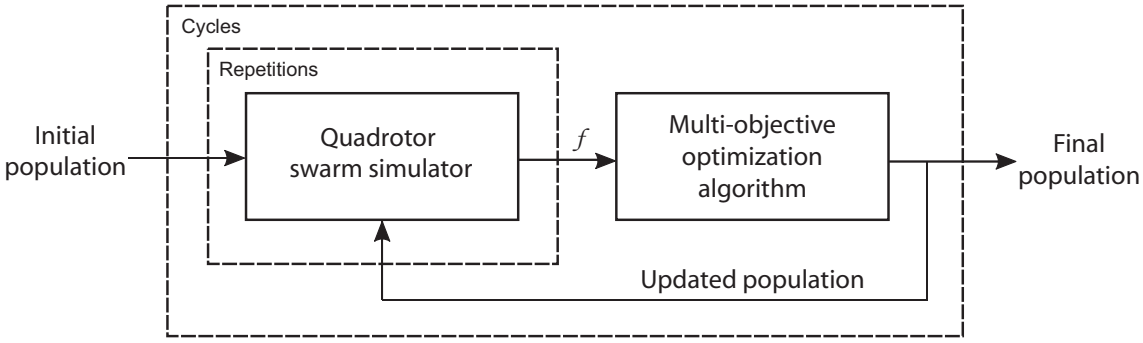


Figure 3.1: Description of experiments 1 to 6

- Experiment 1: Five members without obstacles evaluated by QSS.
- Experiment 2: Ten members without obstacles evaluated by QSS.
- Experiment 3: Twenty members without obstacles evaluated by QSS.
- Experiment 4: Five members with obstacles evaluated by QSS.
- Experiment 5: Ten members with obstacles evaluated by QSS.
- Experiment 6: Twenty members with obstacles evaluated by QSS.

In the Figure 3.1 a diagram that describes the optimization process in the experiments 1 to 6 is showed, where the block of "Multi-objective optimization algorithm" includes the crossover and mutation operators in evolutionary algorithms or velocity and position update in swarm intelligence algorithms.

In this set of experiments the reference point $\boldsymbol{z_{ref}}$ needed for calculate the hypervolume measure explained in the Section 2.5.1.1 is selected using the methodology explained by Knowles [68]. The elements of the vector $\boldsymbol{z_{ref}}$ are selected by:

$$z_{ref,i} = z_{nadir,i} + \delta \left(z_{nadir,i} - z_{ideal,i} \right) \tag{3.1}$$

Where $z_{nadir,i}$ and $z_{ideal,i}$ are the worst and best value in the objective i found by any algorithm in the optimization process. A value of $\delta = 0.1$ is used.

3.1.2 Using nodes-based statistical model

In this set of experiments the multi-objective optimization algorithms estimate the evaluation of the candidate solutions using the proposed surrogate model named nodes-based statistical model (NBSM) explained in the Section 2.3.4, this surrogate model allows the use of larger populations and larger number of cycles or generations, even allows a better comparison of performance between multi-objective optimization algorithms because multiple optimizations per algorithm can be done in a single experiment. However, evaluate the selected nodes is still very expensive in terms of computational time, for that reason, a reduce number of nodes are generated in the search space and each node is evaluated for 10 repetitions in order to obtain a good approximation of its objective functions.

- Experiment 7: Five members without obstacles evaluated by NBSM.
- Experiment 8: Ten members without obstacles evaluated by NBSM.
- Experiment 9: Twenty members without obstacles evaluated by NBSM.
- Experiment 10: Five members with obstacles evaluated by NBSM.
- Experiment 11: Ten members with obstacles evaluated by NBSM.
- Experiment 12: Twenty members with obstacles evaluated by NBSM.

In the Figure 3.2 a diagram that describes the optimization process in the experiments 7 to 12 is showed. Note that the block “Quadrotor swarm simulator” is outside of the optimization cycles, this allows the use of more population and more cycles. However, the evaluation of the nodes, that is an independent process from the optimizations and its information is required by the block “Nodes-based statistical model”, is still very expensive in terms of computational time.

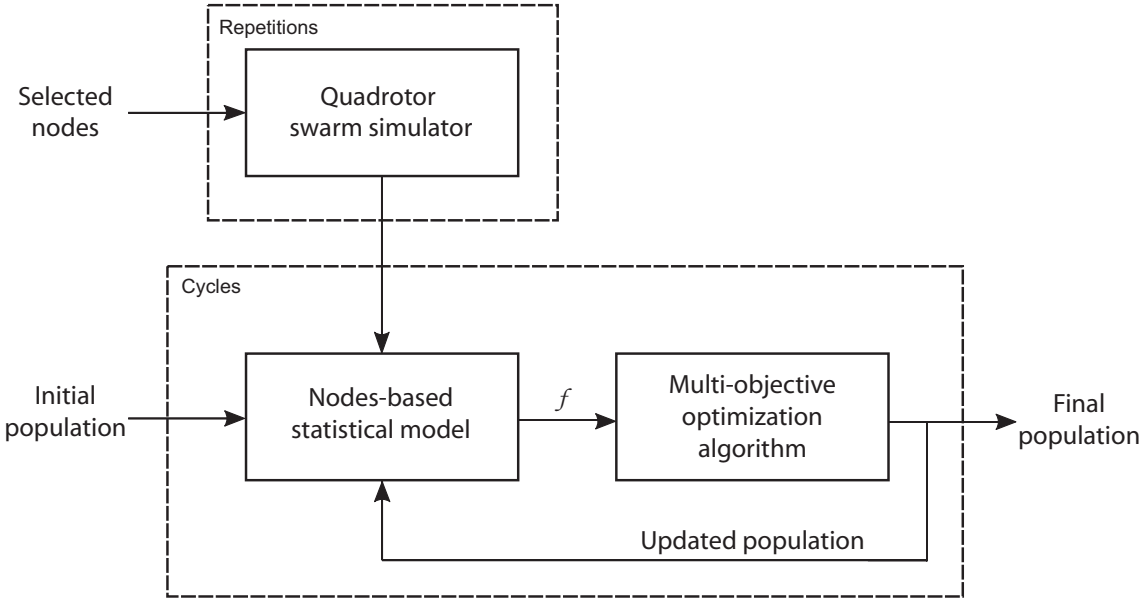


Figure 3.2: Description of experiments 7 to 12

In order to promote a fair comparison between multi-objective optimization algorithms, the selection of the reference point $\boldsymbol{z_{ref}}$ required for calculate the hypervolume measure explained in the Section 2.5.1.1 is done using the methodology proposed by Ishibuchi et al. [69]. The use of this methodology only is possible in the experiments 7 to 12, because in this set of experiments all the multi-objective optimization algorithms produce the same number of optimized solutions at the end of the optimization process. The elements of $\boldsymbol{z_{ref}}$ are selected as follows:

$$z_{ref,i} = 1 + 1/H \tag{3.2}$$

Where the selection of the integer H depends on the optimized solutions found:

$$C_{H+m-1}^{m-1} \leq \mid \boldsymbol{SS} \mid < C_{H+m}^{m-1} \tag{3.3}$$

Where m is the number of objectives. As in all cases 250 optimized solutions were found, a value of $H = 9$ is used. The objective functions are normalized only for the calculation of the hypervolume measure using this methodology.

3.2 General parameters

In this section the parameters for the quadrotor swarm simulator, the nodes-based statistical model and the general optimization parameters are selected. These parameters are the same regardless of the optimization algorithm.

3.2.1 Quadrotor swarm simulator parameters

As the dynamic model and a PID controller are used to simulate the movement of quadrotors, the physical parameters of quadrotors must be defined. Table 3.1 contains these parameters that are estimated using a real prototype of a quadrotor.

Table 3.1: Physical parameters for each quadrotor

Parameter	Description	Value	Units
m	Mass	0.42	kg
g	Gravity	9.81	m/s^2
l	Arm length	0.117	m
a_d	Friction coefficient	0.25	$kg \cdot s$
k	Thrust coefficient	$8.5 \times 10^{-}7$	$N \cdot s^2$
d	Drag coefficient	$1.46 \times 10^{-}8$	$N \cdot m \cdot s^2$
I_{xx}	Inertia on x axis	$7.92 \times 10^{-}6$	$kg \cdot m^2$
I_{yy}	Inertia on y axis	$7.92 \times 10^{-}6$	$kg \cdot m^2$
I_{zz}	Inertia on z axis	$1.56 \times 10^{-}5$	$kg \cdot m^2$

The parameters used for the quadrotor swarm simulator from the Section 2.2.3 are showed in the Table 3.2.

Table 3.2: Simulator parameters

Parameter	Description	Value	Units
t_{max}	Final time	600	s
Δt	Time step interval	0.05	s
P_{range}	Range for obstacles	2.5	m
Δi	Radius of ZOI	10	m
z_d	Desired altitude	2	m
$\phi_{d,speed}$	ϕ to constant speed	0.00	rad
$\theta_{d,speed}$	θ to constant speed	0.03	rad

Through various tests, the selected weights for a safer repulsion behavior that are used in the equation (2.20) are $w_1 = 0.400, w_2 = 0.075, w_3 = 0.025$. The gains for the PID controller explained in the Section 2.2.2.2 are $k_p = 30.0, k_i = 0.5, k_d = 6.0$ and for the PD controller explained in the Section 2.2.2.3 are $k_{P,p} = 1.75, k_{P,d} = 0.65$.

3.2.2 Nodes-based statistical model parameters

Table 3.3 contains the parameters used in the nodes-based statistical model from the Section 2.3.4. The lower and upper bounds are showed in meters.

Table 3.3: Nodes-based statistical model parameters

Parameter	Description	Value
n	Total dimensions	3
$N_{repetitions}$	Total evaluations per node	10
N	Nodes per dimension	$[5, 5, 5]$
lb	Lower bounds	$[0.36, 0.00, 0.00]$
ub	Upper bounds	$[5.00, 5.00, 5.00]$
τ	Temperature	0.1

3.2.3 Optimization parameters

Table 3.4 shows the considerations that were taken in all algorithms. In all te experiments the search space is $[0.36, 5] \times [0, 5]^2$.

Table 3.4: Optimization parameters

Parameter	Description	Exp. 1-6	Exp. 7-12
TI	Total of individuals	10	220
TI_{rep}	Size of repository	100	220
T_{cycles}	Total of generations/cycles	10	250
TR	Dimension of every individual	3	3
$T_{repetitions}$	Total evaluations per candidate solution	10	1
T_{runs}	Optimizations per algorithm	1	10

Where the columns “Exp. 1-6” and “Exp. 7-12” contains the values of the parameters used in the experiments 1 to 6 and 7 to 12 respectively.

3.3 Algorithms parameters

The performance of a given optimization algorithm in a specific problem is sometimes highly dependent of its correct parametrization. Therefore, in order to make a fair comparison the following methodology to select the parameters for the optimization algorithms is adopted. Each algorithm is used multiple times with different set of parameters in the optimization of the experiment 1, this process is repeated in order to find the set of parameters that produces the best performance of the algorithm. The selected set of parameters in the experiment 1 is also used in the experiments 2 to 6. The same process is used in the experiment 7 and the selected parameters in each algorithm remain the same in the experiments 8 to 12. With this methodology all the algorithms are manually parametrized in the experiments 1 and 7, and the selected parameters are also used in the others experiments.

3.3.1 MOPSO parameters

Table 3.5 shows the parameters for MOPSO explained in Section 2.4.2.

Table 3.5: MOPSO parameters

Parameter	Description	Exp. 1-6	Exp. 7-12
w	Inertial weight	0.5	0.75
s	Damping rate	0.025	0.001
β_1	Personal learning coefficient	1.5	1.75
β_2	Global learning coefficient	1.5	1.75
n_{grid}	Number of grids per dimension	5	5
α	Inflation rate	0.1	0.1
β	Leader selection pressure	2	3
γ	Deletion selection pressure	2	3
μ	Mutation rate	0.5	0.5

3.3.2 NSGA-II-DE parameters

Table 3.6 shows the parameters for NSGA-II-DE explained in Section 2.4.3.

Table 3.6: NSGA-II-DE parameters

Parameter	Description	Exp. 1-6	Exp. 7-12
CR	Crossover probability	1	0.9
F	Scaling factor	0.5	0.5
η	Distribution index	20	30
pm	Mutation rate	0.33	0.33

The small amount of parameters used in the NSGA-II-DE is a very good characteristic of this algorithm, because it requires less effort to parametrize.

3.3.3 MOEA/D-DE parameters

Table 3.7 shows the parameters for MOEAD/D-DE explained in Section 2.4.4.

Table 3.7: MOEA/D-DE parameters

Parameter	Description	Exp. 7-12
H	Integer to generate weight vectors	9
T	Total of neighbors	20
nr	Max. solutions replaced by a child	5
δ	Select parents from neighborhood	0.9
CR	Crossover probability	0.9
F	Scaling factor	0.5
η	Distribution index	30
pm	Mutation rate	0.33
n_{grid}	Number of grids per dimension	5
α	Inflation rate	0.1
γ	Deletion selection pressure	3

The MOEA/D-DE is only used in the experiments 7 to 12 because the experiments 1 to 6 are only optimized by MOPSO and NSGA-II-DE due to the highly computational time required by these experiments.

Chapter 4

Results

4.1 Using quadrotor swarm simulator

The results of the experiments 1 to 6 that are optimized using the quadrotor swarm simulator as evaluation function are presented in this section. The obtained results are organized as follows for each experiment: 1) All the nondominated solutions obtained by MOPSO and NSGA-II-DE in the repository at the end of the optimization process are showed, these solutions represent the optimized control parameters and its objective functions. A solution without cooperation between members of the swarm is included in order to compare the quality of the optimized solutions. 2) The Pareto fronts and solution spaces are created using the previous nondominated solutions. 3) The comparison between MOPSO and NSGA-II-DE is given using the selected performance metrics. 4) A study based on bootstrap methods is made using a selected nondominated solution, this is done with the purpose to verify how well 10 evaluations per solution approximates the real objective functions of the solution. 5) At the end of each experiment, a snapshot of the swarm of quadrotors during the simulation of the task using the control parameters of the previous selected solution is showed.

4.1.1 Experiment 1

The optimized control parameters in the experiment of five quadrotors without obstacles can be seen in the Table 4.1 and Table 4.2 for MOPSO and NSGA-II-DE respectively. For comparison purposes, a solution ($\Delta r = 2, \Delta o = 0, \Delta a = 0$) without cooperation between members is studied, the following objective functions are obtained: ($f_1 = 238.1150, f_2 = 11.2639, f_3 = 0.0, f_4 = 0.06$).

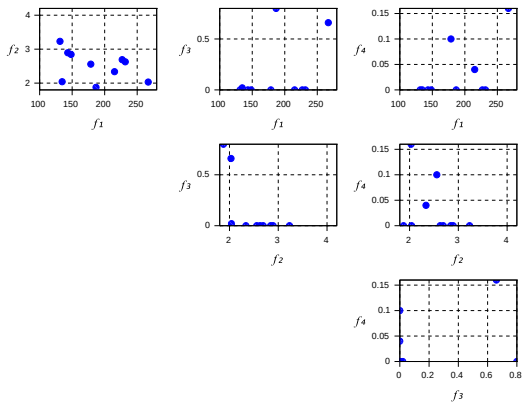
Table 4.1: Nondominated solutions of experiment 1 with MOPSO

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
$MOPSO_1$	0.9754	2.7073	5.0000	143.2440	2.8939	0.0000	0.0000
$MOPSO_2$	1.2192	3.7778	4.0145	131.2990	3.2304	0.0000	0.0000
$MOPSO_3$	0.4202	3.1465	5.0000	266.9620	2.0312	0.6600	0.1600
$MOPSO_4$	0.5824	3.4710	5.0000	215.1500	2.3368	0.0000	0.0400
$MOPSO_5$	0.8758	3.2637	4.0795	148.3430	2.8441	0.0000	0.0000
$MOPSO_6$	0.3600	3.0124	4.8246	186.7060	1.8763	0.8000	0.0000
$MOPSO_7$	0.6353	3.1268	4.3276	134.6100	2.0410	0.0200	0.0000
$MOPSO_8$	0.9947	3.5355	5.0000	178.6820	2.5584	0.0000	0.1000
$MOPSO_9$	1.1539	3.5472	4.9131	231.6790	2.6271	0.0000	0.0000
$MOPSO_{10}$	0.7991	3.0633	4.8938	226.9310	2.6906	0.0000	0.0000

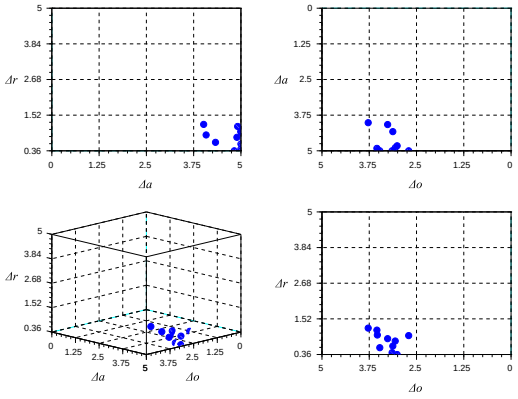
Table 4.2: Nondominated solutions of experiment 1 with NSGA-II-DE

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
NSGA-II-DE_1	2.5076	2.5832	4.0765	155.1220	4.0350	0.0000	0.0000
NSGA-II-DE_2	2.2783	2.8728	3.8903	153.9550	4.0021	0.0000	0.0200
NSGA-II-DE_3	0.8114	3.2898	3.8639	249.0180	2.7465	0.0000	0.0000
NSGA-II-DE_4	0.3816	4.1359	3.1949	111.7810	3.1034	0.6000	0.0000
NSGA-II-DE_5	0.4838	3.5780	3.5274	132.8620	2.3551	0.2800	0.0000
NSGA-II-DE_6	0.8402	3.5567	3.5674	159.3440	3.0517	0.0000	0.0000
NSGA-II-DE_7	0.6589	3.8059	3.4060	227.6450	2.6094	0.0200	0.0000
NSGA-II-DE_8	0.6771	3.7029	3.5870	137.3010	3.2908	0.0200	0.0000
NSGA-II-DE_9	0.9367	3.5178	3.8166	185.8210	3.0293	0.0000	0.0000

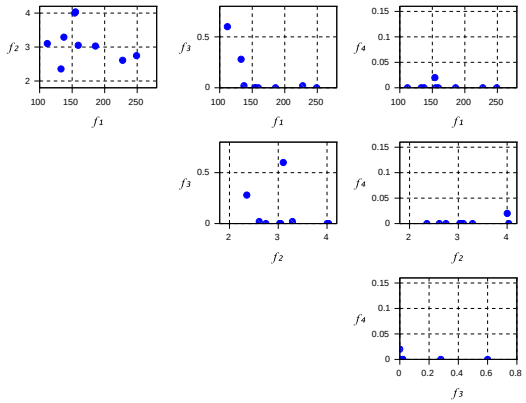
The results form the Table 4.1 and Table 4.2 can be represented by the Pareto fronts and solution spaces showed in the Figure 4.1.



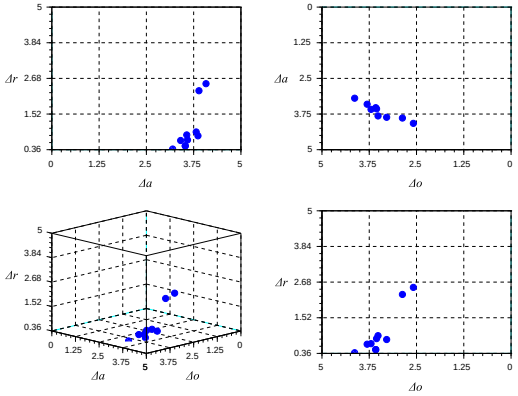
(a) MOPSO: Pareto fronts



(b) MOPSO: Solution space



(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space

Figure 4.1: Pareto fronts and solution spaces in experiment 1

Table 4.3 shows the comparison of the performance in the optimization of the task between MOPSO and NSGA-II-DE in the selected performance metrics. The size of the resulting repository is also included.

Table 4.3: Performance of optimization algorithms in experiment 1

Algorithm	$\mid \boldsymbol{SS} \mid$	HV	Q	TPC
MOPSO	10	0.8131	1.0000	3792.2276
NSGA-II-DE	9	0.6465	0.2222	3796.0067

The solution $MOPSO_1$ is studied more deeply in order to determine the confidence intervals of its objective functions, and verify the veracity of the mean of 10 runs used during the optimization process as shown in the Table 4.4.

Table 4.4: Confidence intervals for solution $MOPSO_1$ in experiment 1

Objectives	10 Runs	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	143.2440	161.1777	161.1769	19.4764	$[121.3485, 201.0069]$
f_2	2.8939	2.8657	2.8675	0.3812	$[2.0861, 3.6453]$
f_3	0.0000	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0000	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$

A snapshot of the swarm behavior during the simulation using the control parameters from the solution $MOPSO_1$ is shown in the Figure 4.2.

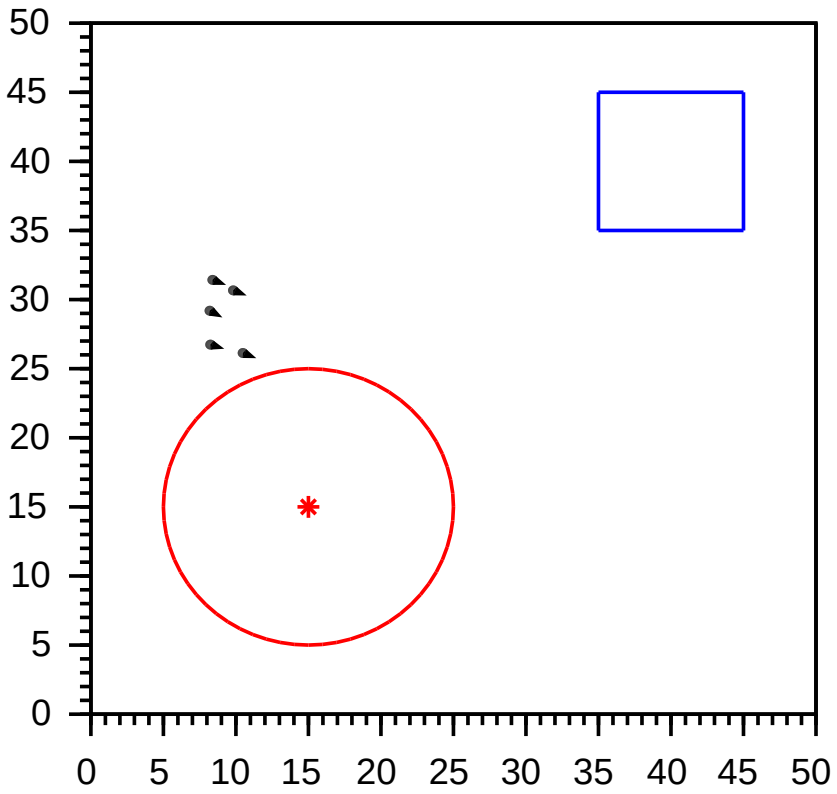


Figure 4.2: Swarm behavior using solution $MOPSO_1$ in experiment 1

4.1.2 Experiment 2

The optimized control parameters in the experiment of ten quadrotors without obstacles can be seen in the Table 4.5 and Table 4.6 for MOPSO and NSGA-II-DE respectively. For comparison purposes, a solution ($\Delta r = 2, \Delta o = 0, \Delta a = 0$) without cooperation between members is studied, the following objective functions are obtained: ($f_1 = 284.8615, f_2 = 11.6314, f_3 = 0.0, f_4 = 0.12$).

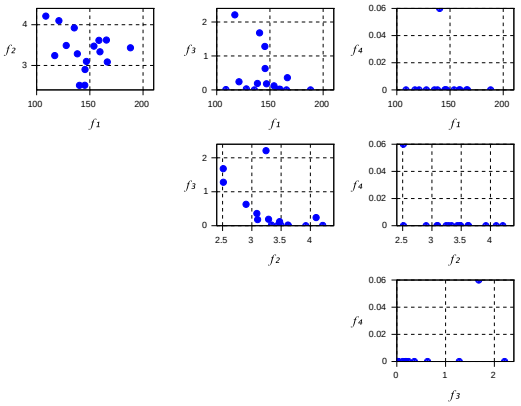
Table 4.5: Nondominated solutions of experiment 2 with MOPSO

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
$MOPSO_1$	0.6827	5.0000	4.4495	127.8050	3.4902	0.0300	0.0000
$MOPSO_2$	0.4567	2.9705	4.6520	117.0130	3.2423	2.2100	0.0000
$MOPSO_3$	0.3600	3.8945	4.2741	140.2550	2.5114	1.6800	0.0600
$MOPSO_4$	0.5757	4.5656	4.6454	146.7640	3.0984	0.1800	0.0000
$MOPSO_5$	0.6902	5.0000	4.4549	153.8300	3.4742	0.1200	0.0000
$MOPSO_6$	0.5287	5.0000	4.0704	166.3990	3.0858	0.3600	0.0000
$MOPSO_7$	0.6424	4.3958	4.2967	138.2555	3.2856	0.1900	0.0000
$MOPSO_8$	0.3600	4.1302	4.5728	145.2215	2.5135	1.2800	0.0000
$MOPSO_9$	0.4182	4.6200	3.6694	145.4630	2.9011	0.6300	0.0000
$MOPSO_{10}$	0.8591	5.0000	3.9501	108.6270	4.2113	0.0100	0.0000
$MOPSO_{11}$	0.8444	4.7738	3.9107	159.5135	3.3355	0.0200	0.0000
$MOPSO_{12}$	0.7507	4.6416	3.8412	158.5355	3.6166	0.0200	0.0000
$MOPSO_{13}$	0.8765	4.4329	4.0778	188.1620	3.4338	0.0000	0.0000
$MOPSO_{14}$	1.0465	4.8252	3.9839	135.3580	3.9218	0.0000	0.0000
$MOPSO_{15}$	0.9597	5.0000	4.3150	165.5700	3.6237	0.0000	0.0000
$MOPSO_{16}$	0.5739	4.9219	4.6058	120.9270	4.0984	0.2400	0.0000

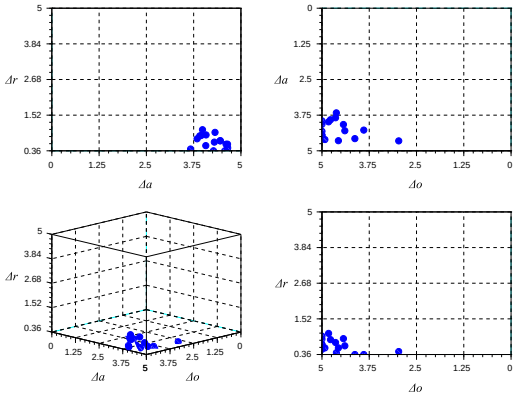
Table 4.6: Nondominated solutions of experiment 2 with NSGA-II-DE

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
NSGA-II-DE_1	0.9324	4.9051	4.5290	155.4425	3.5884	0.0000	0.0000
NSGA-II-DE_2	1.0098	4.4121	3.6188	128.9175	4.2395	0.0000	0.0000
NSGA-II-DE_3	1.3017	4.8943	3.3501	137.7400	4.1922	0.0000	0.0200
NSGA-II-DE_4	0.8007	4.3682	4.6729	202.1020	3.1428	0.0200	0.0000
NSGA-II-DE_5	0.5484	3.8205	3.6379	173.0840	2.7821	0.3900	0.0000
NSGA-II-DE_6	0.6777	3.9738	3.9447	107.0580	3.0299	0.0500	0.0000
NSGA-II-DE_7	0.8604	4.2807	3.3406	151.4605	3.6057	0.0100	0.0000
NSGA-II-DE_8	0.8274	4.2171	3.2833	146.1165	3.3083	0.0300	0.0000
NSGA-II-DE_9	1.1410	4.6971	2.9860	141.8325	3.9632	0.0000	0.0000
NSGA-II-DE_{10}	0.6972	4.4987	4.2327	137.8615	3.7427	0.0400	0.0000
NSGA-II-DE_{11}	0.8133	3.8861	3.0479	130.1580	4.1396	0.0300	0.0000

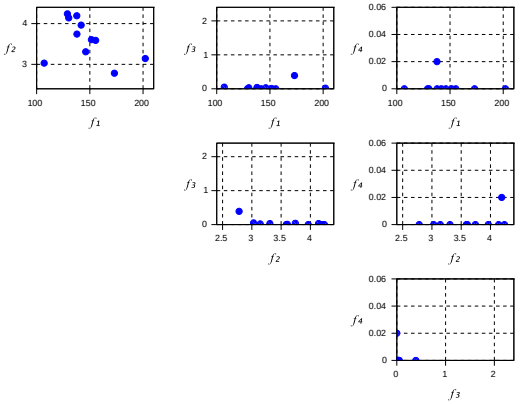
The results form the Table 4.5 and Table 4.6 can be represented by the Pareto fronts and solution spaces showed in the Figure 4.3.



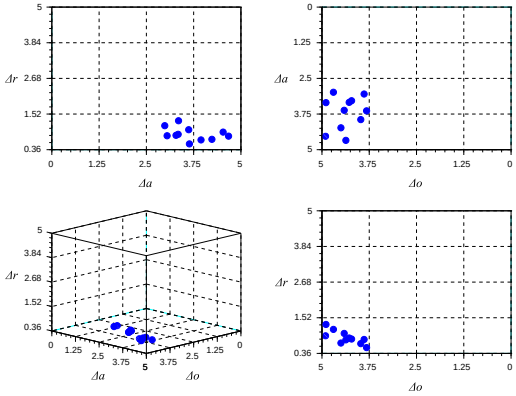
(a) MOPSO: Pareto fronts



(b) MOPSO: Solution space



(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space

Figure 4.3: Pareto fronts and solution spaces in experiment 2

Table 4.7 shows the comparison of the performance in the optimization of the task between MOPSO and NSGA-II-DE in the selected performance metrics. The size of the resulting repository is also included.

Table 4.7: Performance of optimization algorithms in experiment 2

Algorithm	$\mid \boldsymbol{SS} \mid$	HV	Q	TPC
MOPSO	16	0.6573	0.5000	6319.4297
NSGA-II-DE	11	0.7595	0.6364	8561.1657

The solution $MOPSO_{14}$ is studied more deeply in order to determine the confidence intervals of its objective functions, and verify the veracity of the mean of 10 runs used during the optimization process as shown in theTable 4.8.

Table 4.8: Confidence intervals for solution $MOPSO_{14}$ in experiment 2

Objectives	10 Runs	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	135.3580	164.6473	164.7483	15.9515	$[132.0266, 197.2681]$
f_2	3.9218	4.4055	4.4047	0.3220	$[3.7471, 5.0639]$
f_3	0.0000	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0000	0.0067	0.0067	0.0066	$[-0.0068, 0.0202]$

A snapshot of the swarm behavior during the simulation using the control parameters from the solution $MOPSO_{14}$ is shown in the Figure 4.4.

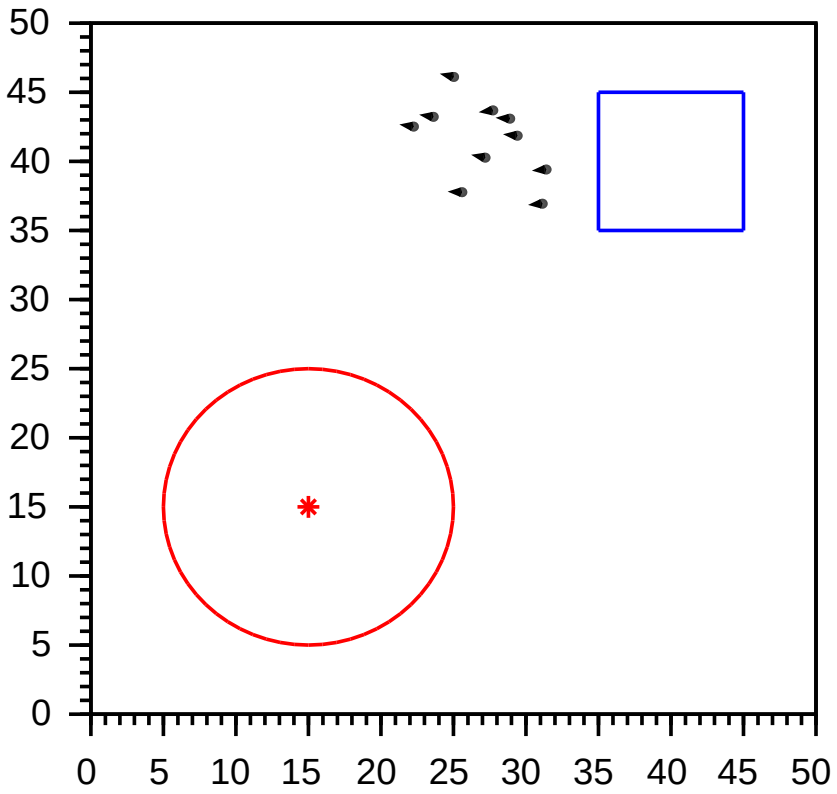


Figure 4.4: Swarm behavior using solution $MOPSO_{14}$ in experiment 2

4.1.3 Experiment 3

The optimized control parameters in the experiment of twenty quadrotors without obstacles can be seen in the Table 4.9 and Table 4.10 for MOPSO and NSGA-II-DE respectively. For comparison purposes, a solution ($\Delta r = 2, \Delta o = 0, \Delta a = 0$) without cooperation between members is studied, the following objective functions are obtained: ($f_1 = 291.5730, f_2 = 11.5034, f_3 = 0.0, f_4 = 0.125$).

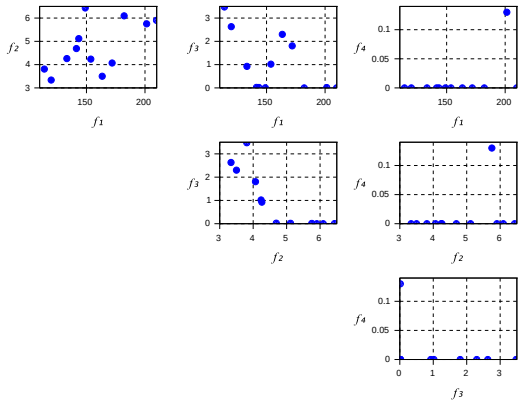
Table 4.9: Nondominated solutions of experiment 3 with MOPSO

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
$MOPSO_1$	0.3600	5.0000	3.9115	114.0078	3.8092	3.4750	0.0000
$MOPSO_2$	0.3600	5.0000	3.9720	119.8583	3.3377	2.6300	0.0000
$MOPSO_3$	1.2696	4.5400	4.5545	148.9490	6.4321	0.0050	0.0000
$MOPSO_4$	0.4144	4.7658	4.1394	163.4595	3.5005	2.3000	0.0000
$MOPSO_5$	0.6928	4.8362	4.6942	133.2480	4.2600	0.9250	0.0000
$MOPSO_6$	0.6046	4.6913	4.6432	153.7175	4.2365	1.0200	0.0000
$MOPSO_7$	1.4181	4.9213	4.0628	182.1735	6.0948	0.0000	0.0000
$MOPSO_8$	0.9886	4.7388	4.8775	141.4215	4.6903	0.0250	0.0000
$MOPSO_9$	1.1673	4.7741	4.0089	143.4348	5.1165	0.0200	0.0000
$MOPSO_{10}$	0.4688	4.7164	3.8203	171.9140	4.0693	1.8050	0.0000
$MOPSO_{11}$	1.3823	5.0000	4.7549	209.8230	5.9028	0.0000	0.0000
$MOPSO_{12}$	1.0982	5.0000	4.9271	201.4218	5.7524	0.0150	0.1300

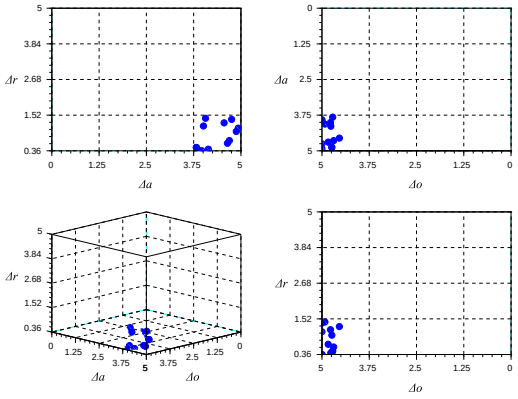
Table 4.10: Nondominated solutions of experiment 3 with NSGA-II-DE

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
NSGA-II-DE_1	1.0818	4.6097	4.9922	167.8208	4.9717	0.0250	0.0000
NSGA-II-DE_2	0.9654	4.4193	3.1257	178.4195	4.5773	0.1350	0.0000
NSGA-II-DE_3	1.2771	4.8624	4.6677	140.4780	5.0750	0.0050	0.0000
NSGA-II-DE_4	0.7192	4.6690	4.5206	125.7953	3.6937	0.2300	0.0000
NSGA-II-DE_5	1.5602	4.9541	4.8992	163.7393	5.0327	0.0000	0.0000
NSGA-II-DE_6	0.5256	4.8644	4.3632	119.8365	4.0457	1.3150	0.0000

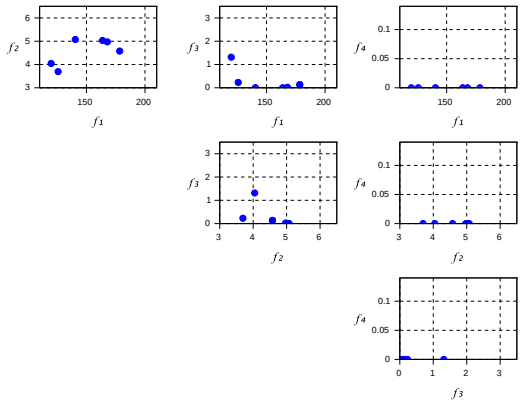
The results form the Table 4.9 and Table 4.10 can be represented by the Pareto fronts and solution spaces showed in the Figure 4.5.



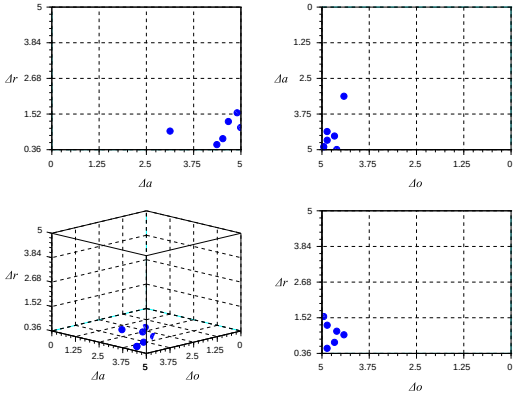
(a) MOPSO: Pareto fronts



(b) MOPSO: Solution space



(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space

Figure 4.5: Pareto fronts and solution spaces in experiment 3

Table 4.11 shows the comparison of the performance in the optimization of the task between MOPSO and NSGA-II-DE in the selected performance metrics. The size of the resulting repository is also included.

Table 4.11: Performance of optimization algorithms in experiment 3

Algorithm	$\mid \boldsymbol{SS} \mid$	HV	Q	TPC
MOPSO	12	0.6853	0.3333	22731.3332
NSGA-II-DE	6	0.8007	0.8333	23424.3809

The solution NSGA-II-DE$_5$ is studied more deeply in order to determine the confidence intervals of its objective functions, and verify the veracity of the mean of 10 runs used during the optimization process as shown in the Table 4.12.

Table 4.12: Confidence intervals for solution NSGA-II-DE$_5$ in experiment 3

Objectives	10 Runs	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	163.7393	186.0362	185.8846	16.9327	[151.4089, 220.6635]
f_2	5.0327	5.3349	5.3341	0.1559	[5.0161, 5.6537]
f_3	0.0000	0.0000	0.0000	0.0000	[0.0000, 0.0000]
f_4	0.0000	0.0000	0.0000	0.0000	[0.0000, 0.0000]

A snapshot of the swarm behavior during the simulation using the control parameters from the solution NSGA-II-DE$_5$ is shown in the Figure 4.6.

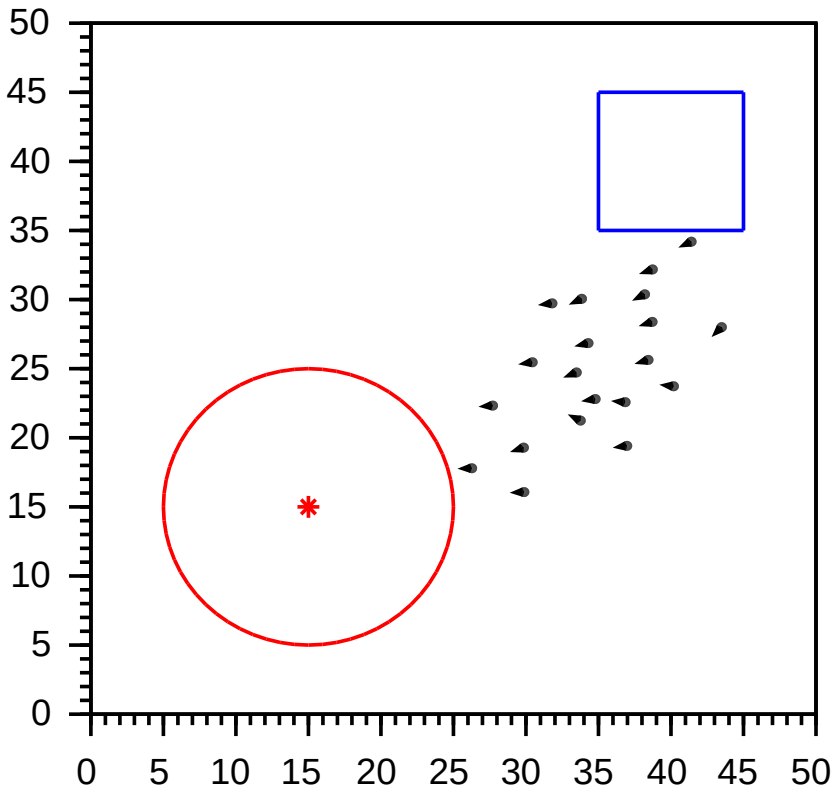


Figure 4.6: Swarm behavior using solution NSGA-II-DE$_5$ in experiment 3

4.1.4 Experiment 4

The optimized control parameters in the experiment of five quadrotors with obstacles can be seen in the Table 4.13 and Table 4.14 for MOPSO and NSGA-II-DE respectively. For comparison purposes, a solution ($\Delta r = 2, \Delta o = 0, \Delta a = 0$) without cooperation between members is studied, the following objective functions are obtained: ($f_1 = 291.7730, f_2 = 12.4994, f_3 = 0.0, f_4 = 0.12$).

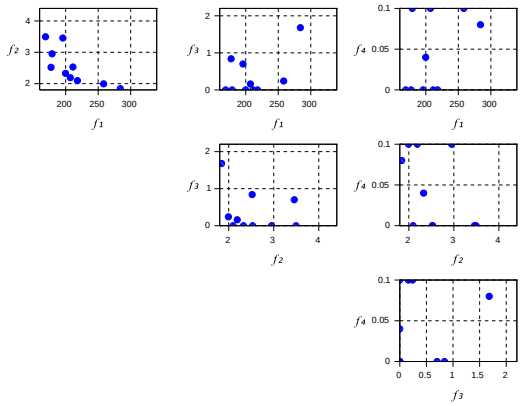
Table 4.13: Nondominated solutions of experiment 4 with MOPSO

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
$MOPSO_1$	0.6776	2.8290	5.0000	218.1140	2.0960	0.0000	0.0000
$MOPSO_2$	1.8216	4.4523	4.8247	169.0730	3.4948	0.0000	0.0000
$MOPSO_3$	0.3600	5.0000	4.7458	195.7070	3.4559	0.7000	0.0000
$MOPSO_4$	1.0199	2.9770	4.1332	199.8200	2.3289	0.0000	0.0400
$MOPSO_5$	0.5467	2.4953	5.0000	258.3700	1.9929	0.2400	0.1000
$MOPSO_6$	1.1093	3.3204	5.0000	211.2180	2.5306	0.0000	0.0000
$MOPSO_7$	0.3600	2.8787	5.0000	284.0340	1.8419	1.6800	0.0800
$MOPSO_8$	0.5865	3.3242	4.8125	207.1620	2.1888	0.1600	0.1000
$MOPSO_9$	0.7385	3.7581	4.8273	179.1480	2.9510	0.0000	0.1000
$MOPSO_{10}$	0.3600	4.1960	4.6375	177.4930	2.5203	0.8400	0.0000

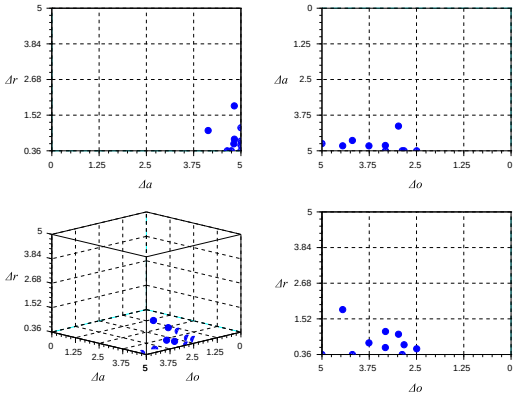
Table 4.14: Nondominated solutions of experiment 4 with NSGA-II-DE

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
NSGA-II-DE_1	1.9735	4.2120	4.4994	228.2610	3.6373	0.0000	0.0000
NSGA-II-DE_2	0.4161	2.3286	4.0644	210.1600	1.8695	0.5400	0.1000
NSGA-II-DE_3	0.3874	2.0358	4.0557	229.4560	2.6637	2.1800	0.0000
NSGA-II-DE_4	1.2401	2.7185	4.6485	321.8740	2.7070	0.0000	0.1000
NSGA-II-DE_5	2.3870	4.9917	4.8699	189.4990	4.2411	0.0000	0.0000
NSGA-II-DE_6	0.8421	3.5249	3.5847	232.2600	2.5438	0.0200	0.0200
NSGA-II-DE_7	0.4260	3.1860	3.3618	233.8630	2.8493	0.9600	0.0000
NSGA-II-DE_8	0.9817	3.6850	3.1730	241.2970	3.2495	0.0000	0.0000
NSGA-II-DE_9	0.9566	3.0953	3.0389	205.5320	2.9128	0.0000	0.0200
NSGA-II-DE_{10}	0.6775	2.5317	3.5429	189.5550	3.3699	0.0200	0.0000
NSGA-II-DE_{11}	2.1232	4.0956	4.8900	226.5660	3.8819	0.0000	0.0000
NSGA-II-DE_{12}	2.2563	4.8902	4.7704	205.9440	3.9275	0.0000	0.0000
NSGA-II-DE_{13}	1.2482	2.7370	4.4014	229.5720	3.2656	0.0000	0.0000

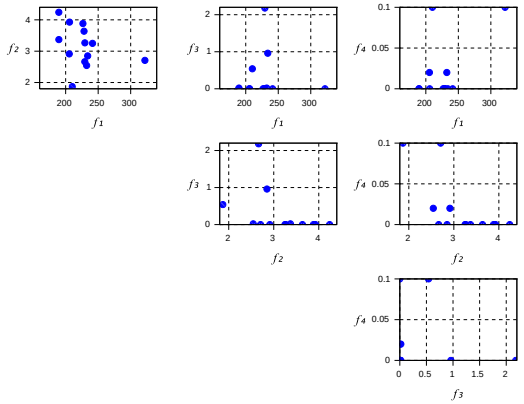
The results form the Table 4.13 and Table 4.14 can be represented by the Pareto fronts and solution spaces showed in the Figure 4.7.



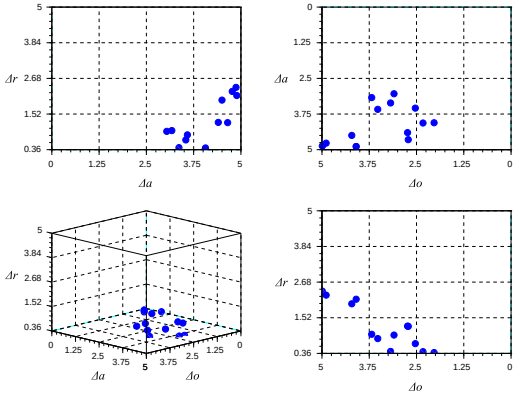
(a) MOPSO: Pareto fronts



(b) MOPSO: Solution space



(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space

Figure 4.7: Pareto fronts and solution spaces in experiment 4

Table 4.15 shows the comparison of the performance in the optimization of the task between MOPSO and NSGA-II-DE in the selected performance metrics. The size of the resulting repository is also included.

Table 4.15: Performance of optimization algorithms in experiment 4

Algorithm	$\mid \boldsymbol{SS} \mid$	HV	Q	TPC
MOPSO	10	0.8277	0.9000	7572.1264
NSGA-II-DE	13	0.5824	0.2308	9744.7240

The solution $MOPSO_6$ is studied more deeply in order to determine the confidence intervals of its objective functions, and verify the veracity of the mean of 10 runs used during the optimization process as shown in the Table 4.16.

Table 4.16: Confidence intervals for solution $MOPSO_6$ in experiment 4

Objectives	10 Runs	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	211.2180	225.1683	225.1684	26.3809	$[171.2193, 279.1174]$
f_2	2.5306	3.0484	3.0440	0.2983	$[2.4382, 3.6585]$
f_3	0.0000	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0000	0.0800	0.0794	0.0470	$[-0.0161, 0.1761]$

A snapshot of the swarm behavior during the simulation using the control parameters from the solution $MOPSO_6$ is shown in the Figure 4.8.

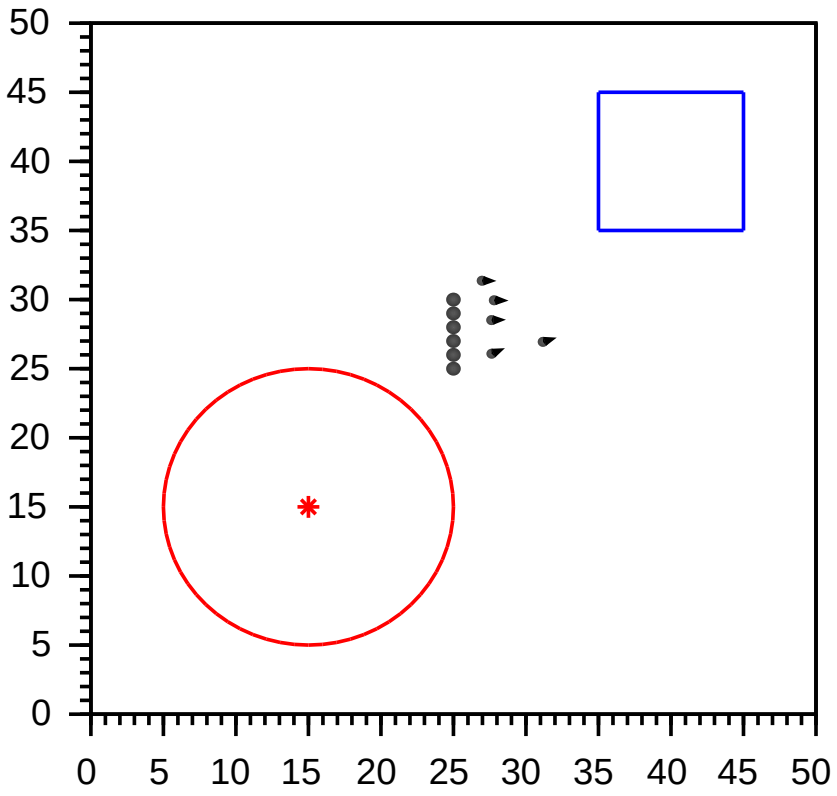


Figure 4.8: Swarm behavior using solution $MOPSO_6$ in experiment 4

4.1.5 Experiment 5

The optimized control parameters in the experiment of ten quadrotors with obstacles can be seen in the Table 4.17 and Table 4.18 for MOPSO and NSGA-II-DE respectively. For comparison purposes, a solution ($\Delta r = 2, \Delta o = 0, \Delta a = 0$) without cooperation between members is studied, the following objective functions are obtained: ($f_1 = 348.3630, f_2 = 12.1598, f_3 = 0.0, f_4 = 0.18$).

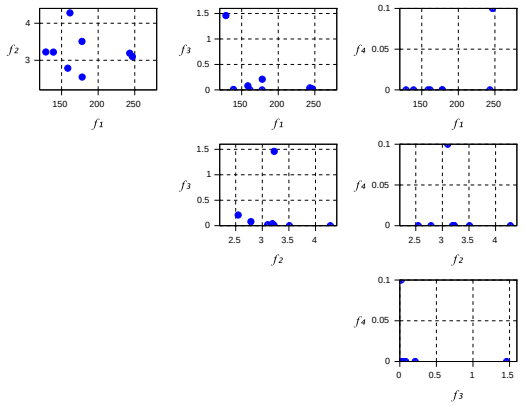
Table 4.17: Nondominated solutions of experiment 5 with MOPSO

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
$MOPSO_1$	0.8506	5.0000	4.0928	177.9010	3.5095	0.0000	0.0000
$MOPSO_2$	0.7062	3.4175	5.0000	158.4860	2.7863	0.0800	0.0000
$MOPSO_3$	0.8461	4.3056	5.0000	246.7410	3.1007	0.0200	0.1000
$MOPSO_4$	1.0151	5.0000	4.8179	161.4245	4.2789	0.0000	0.0000
$MOPSO_5$	0.8226	2.7862	4.6221	243.1530	3.1906	0.0400	0.0000
$MOPSO_6$	0.3600	4.2896	4.8972	128.4525	3.2238	1.4600	0.0000
$MOPSO_7$	0.6589	3.3444	4.2803	178.1140	2.5467	0.2100	0.0000
$MOPSO_8$	0.8226	4.2252	5.0000	138.7860	3.2207	0.0100	0.0000

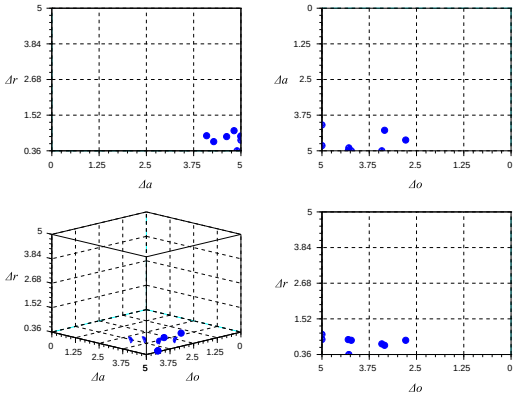
Table 4.18: Nondominated solutions of experiment 5 with NSGA-II-DE

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
NSGA-II-DE_1	0.3804	4.9652	3.4118	164.4815	2.8812	1.2200	0.0000
NSGA-II-DE_2	0.4777	2.5839	4.2065	173.1925	2.2195	0.9500	0.0000
NSGA-II-DE_3	1.0449	4.1707	4.5588	167.3320	3.2652	0.0000	0.0000
NSGA-II-DE_4	1.1089	4.8923	4.3075	148.3185	3.8581	0.0000	0.0000
NSGA-II-DE_5	0.6742	4.9637	3.9764	271.6370	3.2282	0.0400	0.0100
NSGA-II-DE_6	0.5049	4.5516	4.7231	264.6335	2.9796	0.5200	0.1000
NSGA-II-DE_7	0.4574	4.1738	3.6867	204.8455	3.1220	0.7800	0.1000
NSGA-II-DE_8	0.9852	2.5851	4.8113	217.1800	3.1750	0.0100	0.0200
NSGA-II-DE_9	0.4441	3.7845	4.6263	228.8010	2.7270	0.9400	0.0000
NSGA-II-DE_{10}	0.9365	3.7757	4.4139	150.1475	3.6897	0.0000	0.0000
NSGA-II-DE_{11}	0.7058	4.0410	3.9537	239.5215	3.1877	0.1900	0.0000
NSGA-II-DE_{12}	0.5480	4.4362	4.0248	236.6975	3.1882	0.4600	0.0100
NSGA-II-DE_{13}	0.4793	4.7106	4.5354	209.5865	2.9223	0.6800	0.0000
NSGA-II-DE_{14}	0.5423	4.9744	4.3900	230.0990	3.2228	0.2800	0.0000

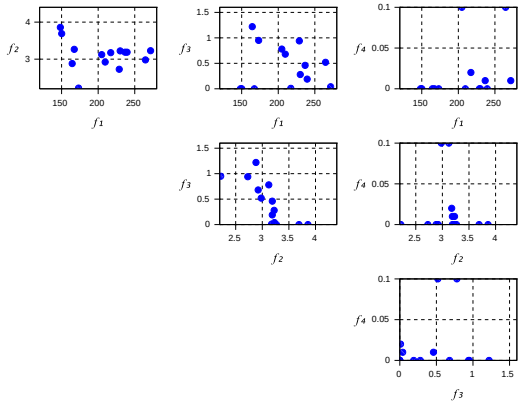
The results form the Table 4.17 and Table 4.18 can be represented by the Pareto fronts and solution spaces showed in the Figure 4.9.



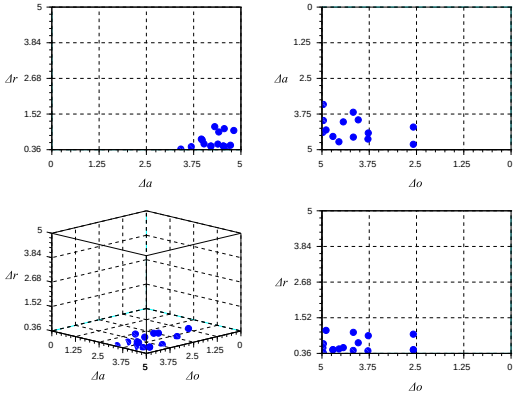
(a) MOPSO: Pareto fronts



(b) MOPSO: Solution space



(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space

Figure 4.9: Pareto fronts and solution spaces in experiment 5

Table 4.19 shows the comparison of the performance in the optimization of the task between MOPSO and NSGA-II-DE in the selected performance metrics. The size of the resulting repository is also included.

Table 4.19: Performance of optimization algorithms in experiment 5

Algorithm	$\mid \boldsymbol{SS} \mid$	HV	Q	TPC
MOPSO	8	0.7339	0.7500	14899.2899
NSGA-II-DE	14	0.6043	0.3571	19666.2177

The solution NSGA-II-DE$_3$ is studied more deeply in order to determine the confidence intervals of its objective functions, and verify the veracity of the mean of 10 runs used during the optimization process as shown in the Table 4.20.

Table 4.20: Confidence intervals for solution NSGA-II-DE$_3$ in experiment 5

Objectives	10 Runs	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	167.3320	186.5690	186.6511	15.5648	$[154.7389, 218.3991]$
f_2	3.2652	3.5005	3.4985	0.1656	$[3.1619, 3.8391]$
f_3	0.0000	0.0033	0.0034	0.0033	$[-0.0034, 0.0100]$
f_4	0.0000	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$

A snapshot of the swarm behavior during the simulation using the control parameters from the solution NSGA-II-DE$_3$ is shown in the Figure 4.10.

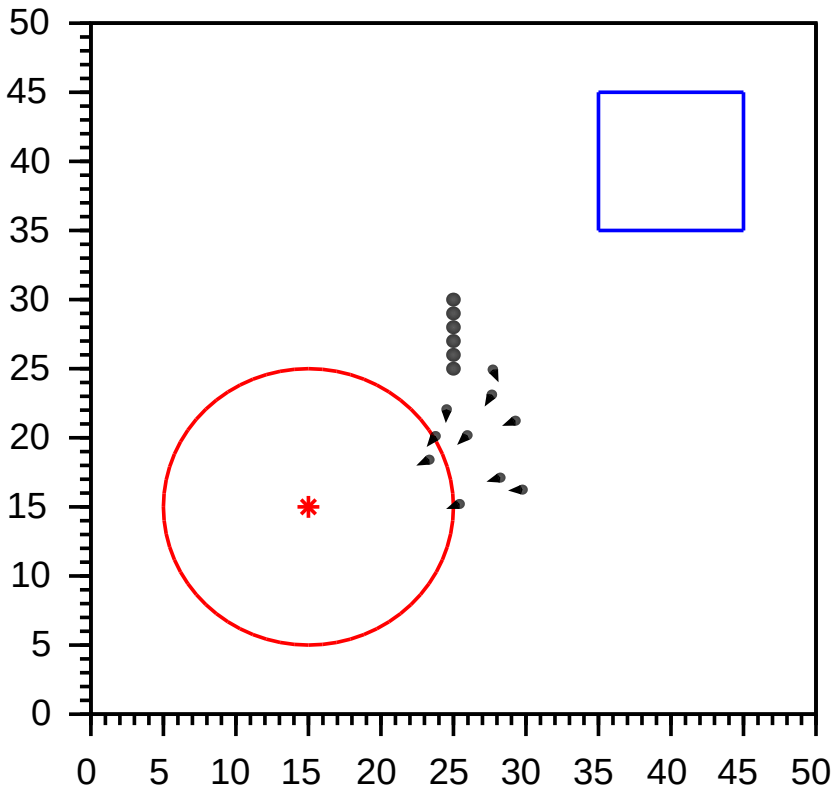


Figure 4.10: Swarm behavior using solution NSGA-II-DE$_3$ in experiment 5

4.1.6 Experiment 6

The optimized control parameters in the experiment of twenty quadrotors with obstacles can be seen in the Table 4.21 and Table 4.22 for MOPSO and NSGA-II-DE respectively. For comparison purposes, a solution ($\Delta r = 2, \Delta o = 0, \Delta a = 0$) without cooperation between members is studied, the following objective functions are obtained: ($f_1 = 324.7952, f_2 = 12.2402, f_3 = 0.0, f_4 = 0.14$).

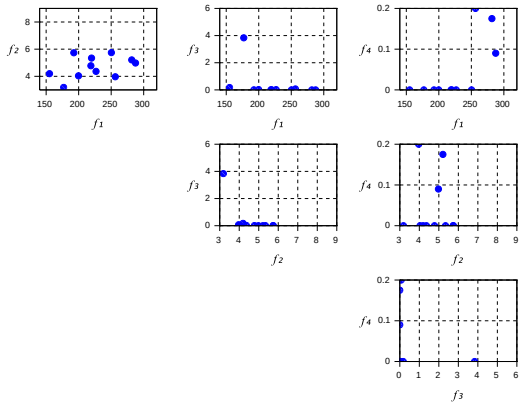
Table 4.21: Nondominated solutions of experiment 6 with MOPSO

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
$MOPSO_1$	0.9468	3.6892	4.9725	226.7110	4.3649	0.0200	0.0000
$MOPSO_2$	1.2379	5.0000	3.7745	281.6193	5.2088	0.0000	0.1750
$MOPSO_3$	0.3600	5.0000	3.9204	176.9430	3.1865	3.8350	0.0000
$MOPSO_4$	0.8330	5.0000	4.4325	256.3508	3.9673	0.0700	0.2000
$MOPSO_5$	1.0950	5.0000	2.6334	219.8010	5.3434	0.0050	0.0000
$MOPSO_6$	1.2011	5.0000	5.0000	287.4393	4.9801	0.0000	0.0900
$MOPSO_7$	1.1684	4.4995	5.0000	250.3205	5.7452	0.0000	0.0000
$MOPSO_8$	1.1454	5.0000	3.8061	192.5420	5.7332	0.0050	0.0000
$MOPSO_9$	0.9729	4.6090	4.1599	199.7828	4.0367	0.0250	0.0000
$MOPSO_{10}$	0.7596	4.7389	4.5322	155.1403	4.1954	0.1750	0.0000
$MOPSO_{11}$	1.0467	5.0000	4.7158	218.8108	4.7781	0.0100	0.0000

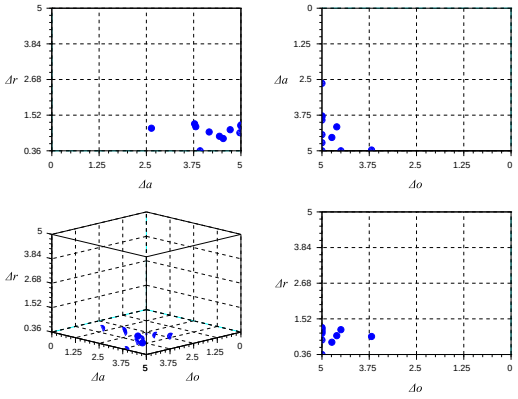
Table 4.22: Nondominated solutions of experiment 6 with NSGA-II-DE

Solution	Δr	Δo	Δa	f_1	f_2	f_3	f_4
NSGA-II-DE_1	0.4045	4.4049	3.1231	232.7063	4.5859	5.0150	0.0000
NSGA-II-DE_2	1.5615	4.3761	2.7090	281.8650	8.2353	0.0000	0.0000
NSGA-II-DE_3	1.3618	4.6860	2.2237	279.3688	8.3452	0.0050	0.0450
NSGA-II-DE_4	1.0313	4.0132	3.1284	312.7245	6.4752	0.0500	0.1350
NSGA-II-DE_5	0.7706	4.7813	2.9562	200.6423	5.6626	0.2800	0.0600
NSGA-II-DE_6	0.6708	4.9363	2.8726	234.1123	4.6704	0.8000	0.1000
NSGA-II-DE_7	0.6015	4.5260	2.1272	250.3075	5.5401	1.6100	0.0300
NSGA-II-DE_8	0.5876	4.2983	2.9122	259.8013	5.0712	2.7800	0.0050
NSGA-II-DE_9	0.4367	4.6685	2.5822	256.6595	4.4307	3.2300	0.0000
NSGA-II-DE_{10}	0.8030	4.6149	2.1604	209.1363	7.3133	0.6300	0.0000
NSGA-II-DE_{11}	0.8785	4.6471	2.3253	211.8570	5.8927	0.2250	0.0000
NSGA-II-DE_{12}	1.1758	4.3195	2.5984	222.5290	6.8189	0.0400	0.0000

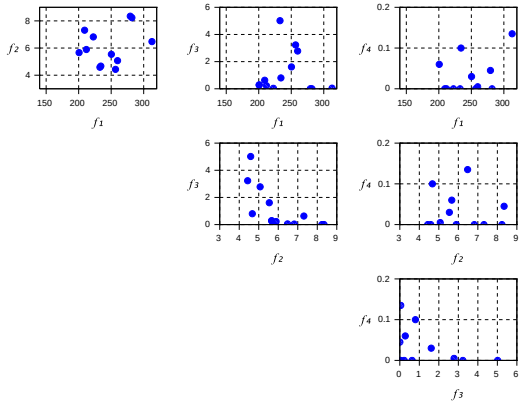
The results form the Table 4.21 and Table 4.22 can be represented by the Pareto fronts and solution spaces showed in the Figure 4.11.



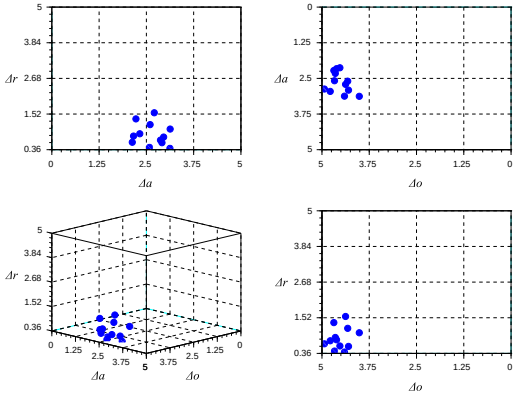
(a) MOPSO: Pareto fronts



(b) MOPSO: Solution space



(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space

Figure 4.11: Pareto fronts and solution spaces in experiment 6

Table 4.23 shows the comparison of the performance in the optimization of the task between MOPSO and NSGA-II-DE in the selected performance metrics. The size of the resulting repository is also included.

Table 4.23: Performance of optimization algorithms in experiment 6

Algorithm	$\mid \boldsymbol{SS} \mid$	HV	Q	TPC
MOPSO	11	0.8776	1.0000	46313.8161
NSGA-II-DE	12	0.4621	0.0000	53620.0161

The solution $MOPSO_7$ is studied more deeply in order to determine the confidence intervals of its objective functions, and verify the veracity of the mean of 10 runs used during the optimization process as shown in the Table 4.24.

Table 4.24: Confidence intervals for solution $MOPSO_7$ in experiment 6

Objectives	10 Runs	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	250.3205	231.2318	231.4261	18.1457	$[194.1238, 268.3399]$
f_2	5.7452	5.3126	5.3136	0.3017	$[4.6956, 5.9295]$
f_3	0.0000	0.0233	0.0231	0.0150	$[-0.0073, 0.5399]$
f_4	0.0000	0.0050	0.0049	0.0049	$[-0.0050, 0.0150]$

A snapshot of the swarm behavior during the simulation using the control parameters from the solution $MOPSO_7$ is shown in the Figure 4.12.

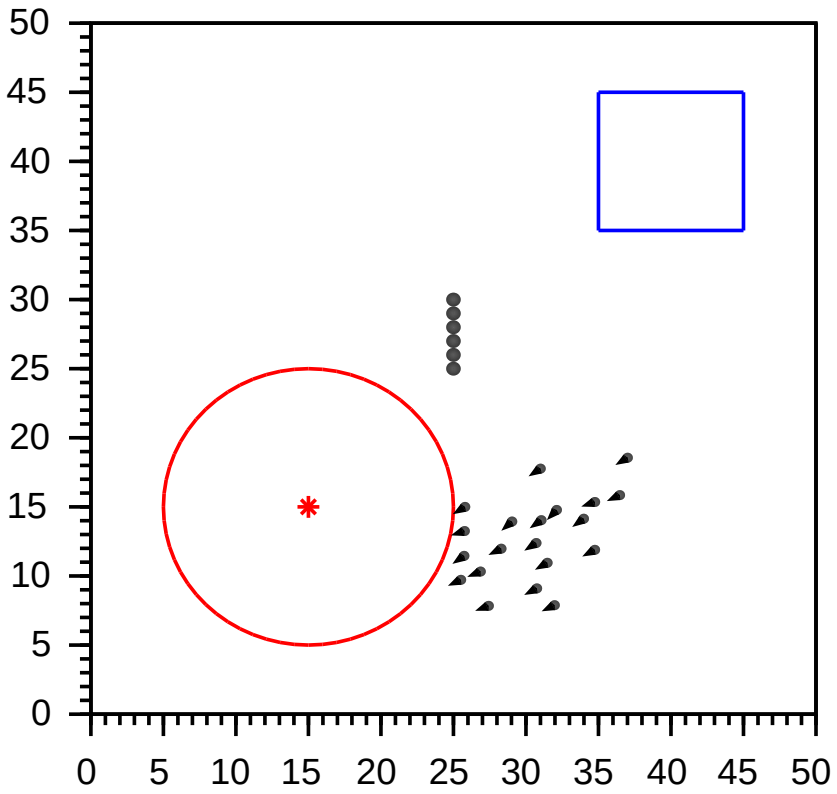


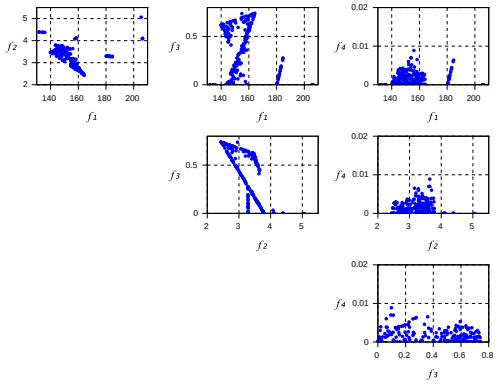
Figure 4.12: Swarm behavior using solution $MOPSO_7$ in experiment 6

4.2 Using nodes-based statistical model

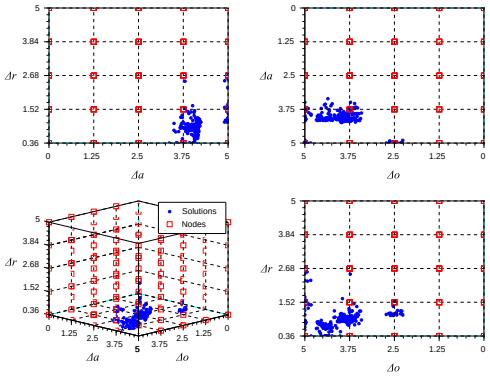
In this section the results of the experiments 7 to 12 that are optimized using the nodes-based statistical model as evaluation function are presented. The same set of results are obtained by each experiment and such results are organized as follows: 1) The multi-objective optimization algorithms MOPSO, NSGA-II-DE and MOEAD/D-DE are used to optimize each experiment for 10 independent optimizations, the best evaluated optimization of each algorithm in terms of HV is selected and its nondominated solutions in the repository at the end of the optimization process are selected to visualize the estimated Pareto fronts and estimated solution spaces. 2) As the surrogate model allows 10 independent optimizations of each experiment per algorithm without high computational cost, a better comparison between optimization algorithms can be performed. Each algorithm is evaluated 10 times using its 10 independent sets of nondominated solutions and the following measures are obtained: average, median, standard deviation, worst value and best value in each performance metric. 3) The relationships between control parameters and objective functions in each experiment can be seen by the generation of 1000 random standardized values from the multivariate normal distribution with mean vector and covariance matrix calculated from the information of the selected nodes, as the nodes are evaluated directly from the simulator, the information from the standardized values is very reliable. Thus, the orientation of the ellipse that contains the generated values for each combination between control parameters and objective functions give important information about its relationship. 4) The best evaluated optimization algorithm in terms of HV in the average measure is selected, then 5 nondominated solutions from the best independent optimization in terms of HV are selected and its confidence intervals using bootstrap methods are calculated. This is done in order to observe how well the nodes-based statistical model estimates the objective functions of the nondominated solutions.

4.2.1 Experiment 7

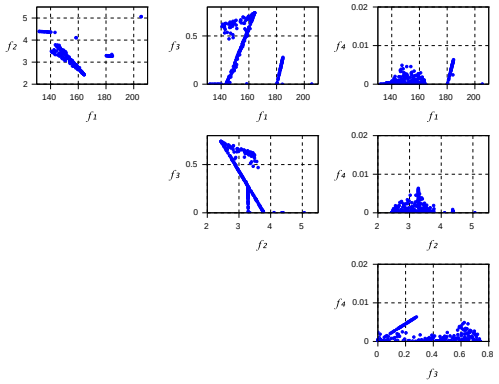
In the Figure 4.13 the estimated Pareto fronts and the solution spaces obtained by each multi-objective optimization algorithm are showed.



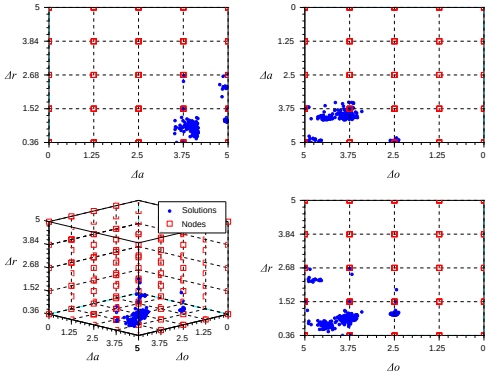
(a) MOPSO: Pareto fronts



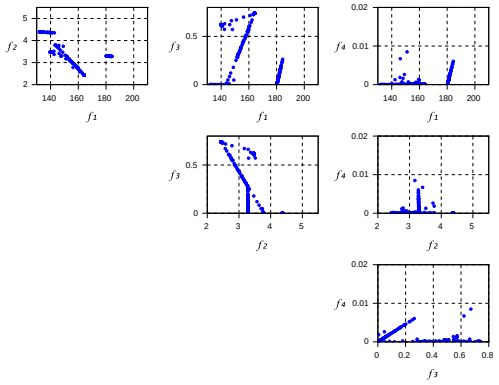
(b) MOPSO: Solution space



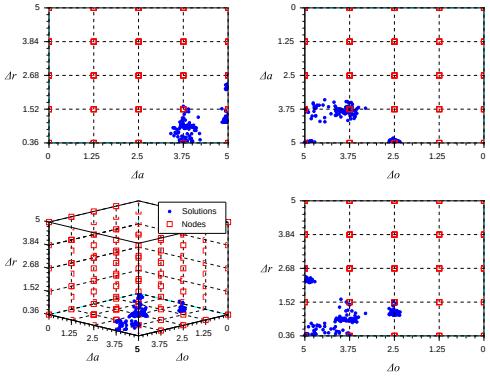
(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space



(e) MOEA/D-DE: Pareto fronts



(f) MOEA/D-DE: Solution space

Figure 4.13: Estimated Pareto fronts and solution spaces in experiment 7

In the Table 4.25 a comparison between multi-objective optimization algorithms after ten independent optimizations using the selected performance metrics is showed.

Table 4.25: Performance of optimization algorithms in experiment 7

Algorithm	MOPSO			NSGA-II-DE			MOEA/D-DE		
Metric	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*
Average	0.7181	0.6736	0.8868	0.7202	0.8755	2.8228	0.7122	0.8441	1.2899
Median	0.7182	0.6818	0.8865	0.7203	0.8750	2.8208	0.7123	0.8477	1.2792
Std. Dev.	0.0009	0.0349	0.0258	0.0005	0.0149	0.0107	0.0030	0.0386	0.0516
Worst	0.7160	0.6182	0.9387	0.7195	0.8500	2.8425	0.7077	0.7818	1.3804
Best	0.7191	0.7364	0.8554	0.7209	0.9000	2.8085	0.7166	0.9045	1.2042

In the Figure 4.14 the relationships between control parameters (Δr, Δo, Δa) and objective functions (f_1, f_2, f_3, f_4) in the experiment of five quadrotors without obstacles can be seen.

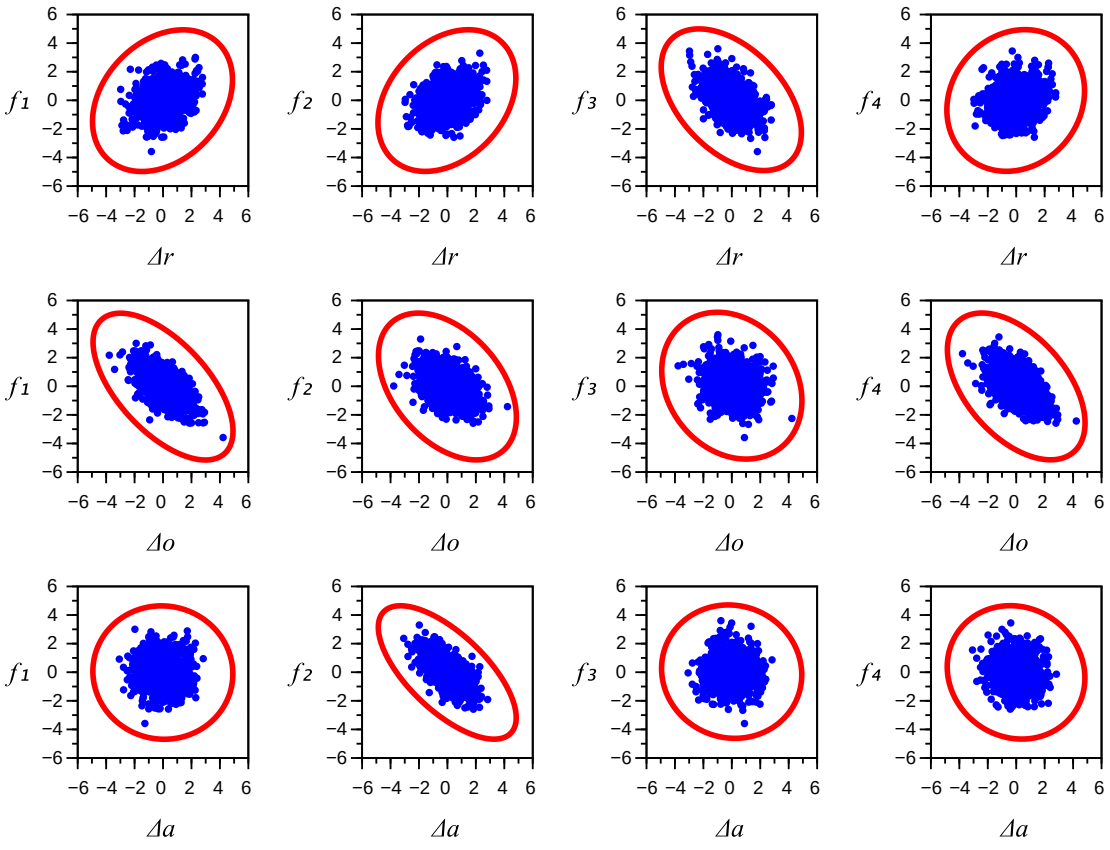


Figure 4.14: Relationship between parameters and functions in experiment 7

Confidence intervals using bootstrap methods are calculated for five random solutions of taken from the best independent optimization in terms of hypervolume measure using NSGA-II-DE as showed in the Table 4.26.

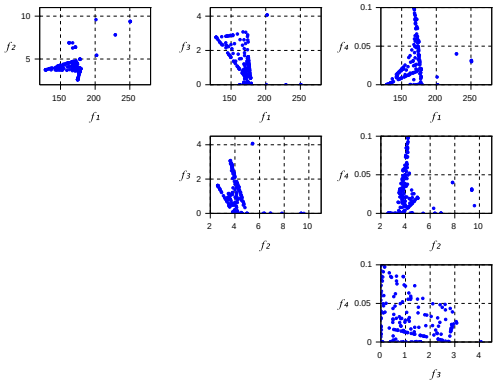
Table 4.26: Confidence intervals in experiment 7

Solution	NSGA-II-DE$_{84}$ ($\Delta_r = 1.0200$, $\Delta_o = 3.9058$, $\Delta_a = 4.0990$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	149.1399	163.7607	163.6921	19.7133	$[123.4470, 204.0744]$
f_2	3.5038	3.9072	3.9043	0.3883	$[3.1130, 4.7013]$
f_3	0.1566	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0013	0.0267	0.0266	0.0207	$[-0.0156, 0.0689]$
Solution	NSGA-II-DE$_{86}$ ($\Delta_r = 1.0947$, $\Delta_o = 2.6112$, $\Delta_a = 4.9248$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	182.8608	194.1833	193.9961	24.6999	$[143.6721, 244.6946]$
f_2	3.2840	3.0940	3.0963	0.4403	$[2.1936, 3.9943]$
f_3	0.1657	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0038	0.0533	0.0529	0.0376	$[-0.0236, 0.1302]$
Solution	NSGA-II-DE$_{154}$ ($\Delta_r = 0.7467$, $\Delta_o = 3.7817$, $\Delta_a = 3.9509$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	163.7128	204.7897	204.8171	23.7527	$[156.2155, 253.3639]$
f_2	2.4659	2.9355	2.9346	0.2291	$[2.4670, 3.4039]$
f_3	0.7209	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0001	0.0533	0.0533	0.0361	$[-0.0206, 0.1272]$
Solution	NSGA-II-DE$_{208}$ ($\Delta_r = 1.0209$, $\Delta_o = 3.7689$, $\Delta_a = 3.9228$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	147.2418	189.6873	189.7062	25.5588	$[137.4197, 241.9550]$
f_2	3.5473	3.7415	3.7426	0.3730	$[2.9786, 4.5043]$
f_3	0.1300	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0001	0.0533	0.0532	0.0336	$[-0.0154, 0.1220]$
Solution	NSGA-II-DE$_{218}$ ($\Delta_r = 0.7813$, $\Delta_o = 4.3142$, $\Delta_a = 3.9924$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	156.8020	152.0193	151.9993	19.1203	$[112.9183, 191.1204]$
f_2	2.8468	2.9166	2.9164	0.0509	$[2.8125, 3.0206]$
f_3	0.6306	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0007	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$

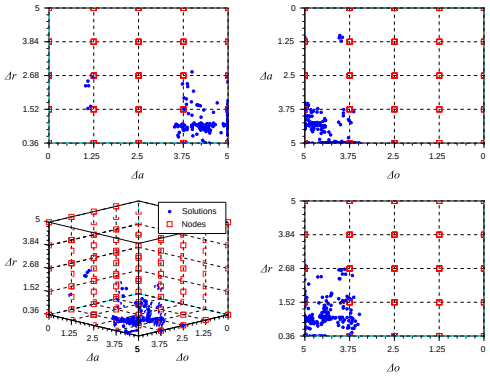
Table 4.26 shows that in experiment 1, 100% of selected solutions has a valid estimation in f_1, 80% of selected solutions has a valid estimation in f_2, 0% of selected solutions has a valid estimation in f_3 and 80% of selected solutions has a valid estimation in f_4.

4.2.2 Experiment 8

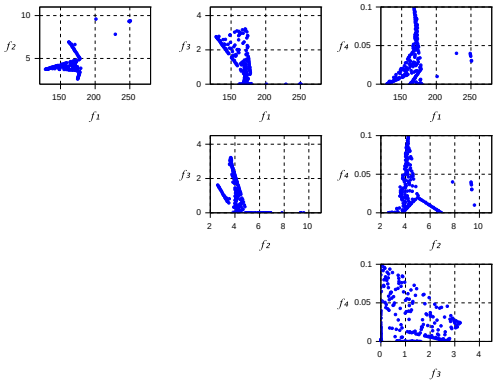
In the Figure 4.15 the estimated Pareto fronts and the solution spaces obtained by each multi-objective optimization algorithm are showed.



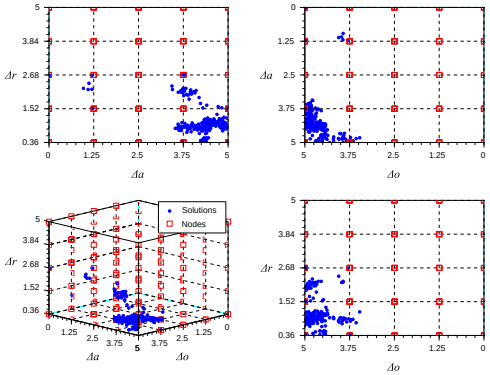
(a) MOPSO: Pareto fronts



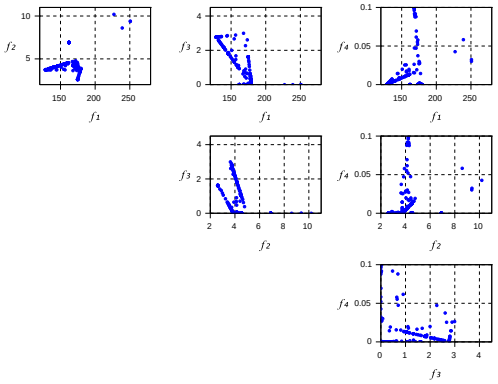
(b) MOPSO: Solution space



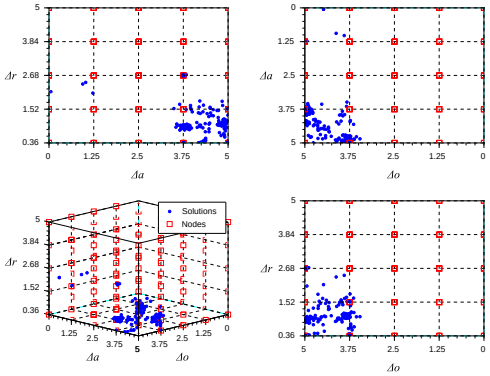
(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space



(e) MOEA/D-DE: Pareto fronts



(f) MOEA/D-DE: Solution space

Figure 4.15: Estimated Pareto fronts and solution spaces in experiment 8

In the Table 4.27 a comparison between multi-objective optimization algorithms after ten independent optimizations using the selected performance metrics is showed.

Table 4.27: Performance of optimization algorithms in experiment 8

Algorithm	MOPSO			NSGA-II-DE			MOEA/D-DE		
Metric	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*
Average	0.8591	0.8991	0.8782	0.8549	0.8723	2.8321	0.8590	0.9395	1.3290
Median	0.8594	0.9045	0.8766	0.8542	0.8750	2.8332	0.8589	0.9432	1.3266
Std. Dev.	0.0014	0.0240	0.0149	0.0037	0.0164	0.0045	0.0012	0.0198	0.0860
Worst	0.8561	0.8545	0.9022	0.8509	0.8455	2.8364	0.8572	0.9045	1.4744
Best	0.8606	0.9318	0.8512	0.8601	0.8955	2.8207	0.8616	0.9591	1.2075

In the Figure 4.16 the relationships between control parameters (Δr, Δo, Δa) and objective functions (f_1, f_2, f_3, f_4) in the experiment of ten quadrotors without obstacles can be seen.

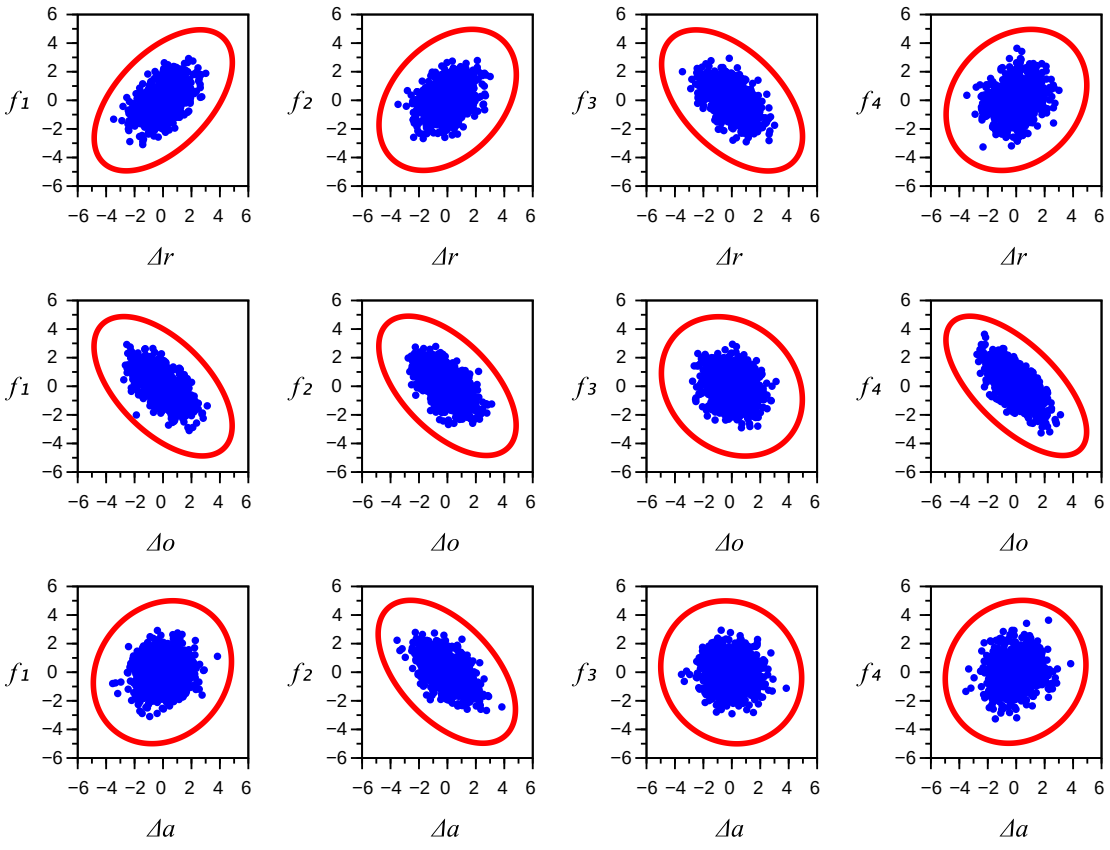


Figure 4.16: Relationship between parameters and functions in experiment 8

Confidence intervals using bootstrap methods are calculated for five random solutions of taken from the best independent optimization in terms of hypervolume measure using MOPSO as showed in the Table 4.28.

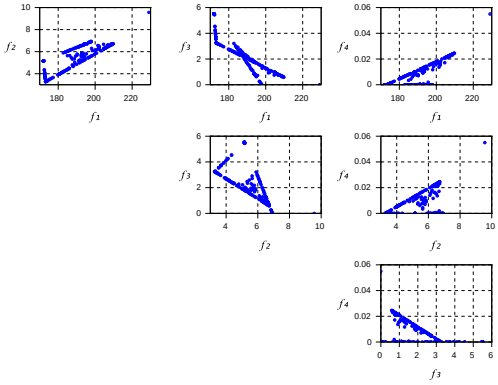
Table 4.28: Confidence intervals in experiment 8

Solution	$MOPSO_{88}$ ($\Delta_r = 1.0272$, $\Delta_o = 4.4319$, $\Delta_a = 5.0000$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	172.8562	160.9753	161.0060	17.2105	$[125.7798, 196.1709]$
f_2	3.9485	3.6701	3.6680	0.1565	$[3.3500, 3.9902]$
f_3	0.5909	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0625	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
Solution	$MOPSO_{97}$ ($\Delta_r = 0.9963$, $\Delta_o = 5.0000$, $\Delta_a = 3.7385$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	166.1181	193.1793	193.1402	20.4609	$[151.3367, 235.0219]$
f_2	4.6670	4.8318	4.8359	0.4251	$[3.9624, 5.7011]$
f_3	0.6791	0.0033	0.0033	0.0032	$[-0.0033, 0.0100]$
f_4	0.0151	0.0433	0.0436	0.0308	$[-0.0196, 0.1062]$
Solution	$MOPSO_{117}$ ($\Delta_r = 0.9871$, $\Delta_o = 4.3467$, $\Delta_a = 5.0000$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	175.5621	155.5820	155.7198	13.0040	$[128.9888, 182.1752]$
f_2	3.6813	3.7222	3.7262	0.2629	$[3.1847, 4.2598]$
f_3	0.7501	0.0033	0.0034	0.0033	$[-0.0035, 0.0101]$
f_4	0.0327	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
Solution	$MOPSO_{141}$ ($\Delta_r = 0.9275$, $\Delta_o = 4.0097$, $\Delta_a = 5.0000$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	176.7162	164.3175	164.1803	20.2483	$[122.9098, 205.7252]$
f_2	3.1714	3.3831	3.3826	0.1613	$[3.0533, 3.7129]$
f_3	0.9069	0.0067	0.0067	0.0046	$[-0.0027, 0.0160]$
f_4	0.0005	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
Solution	$MOPSO_{155}$ ($\Delta_r = 2.3899$, $\Delta_o = 4.6390$, $\Delta_a = 3.7254$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	163.1706	197.9010	197.6924	16.2417	$[164.6868, 231.1152]$
f_2	6.8918	5.2329	5.2318	0.2769	$[4.6666, 5.7992]$
f_3	0.0005	0.0167	0.0167	0.0095	$[-0.0027, 0.0360]$
f_4	0.0002	0.0167	0.0166	0.0163	$[-0.0167, 0.0500]$

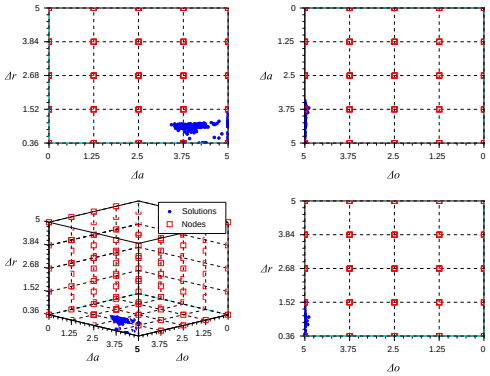
Table 4.28 shows that in experiment 2, 80% of selected solutions has a valid estimation in f_1, 80% of selected solutions has a valid estimation in f_2, 20% of selected solutions has a valid estimation in f_3 and 40% of selected solutions has a valid estimation in f_4.

4.2.3 Experiment 9

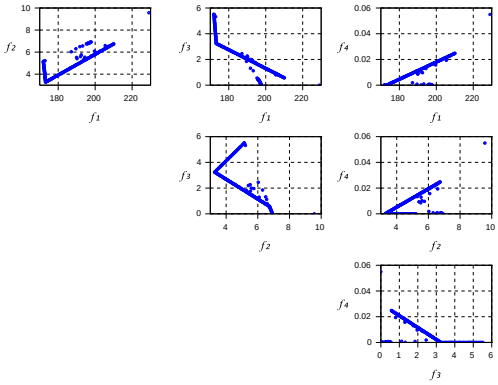
In the Figure 4.17 the estimated Pareto fronts and the solution spaces obtained by each multi-objective optimization algorithm are showed.



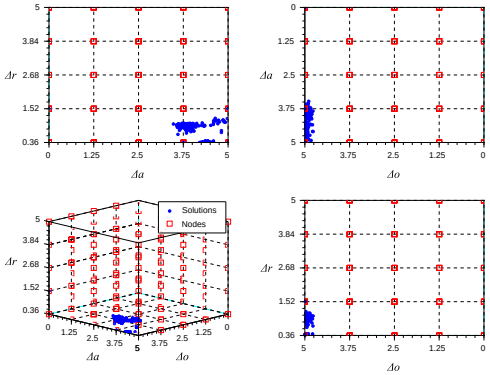
(a) MOPSO: Pareto fronts



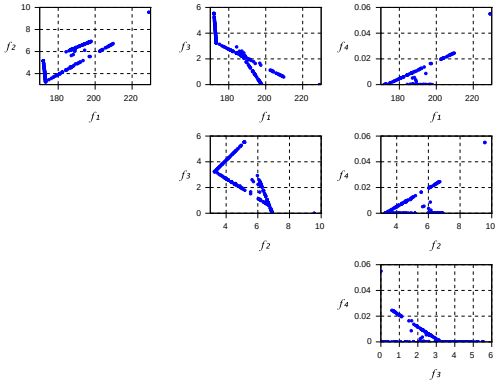
(b) MOPSO: Solution space



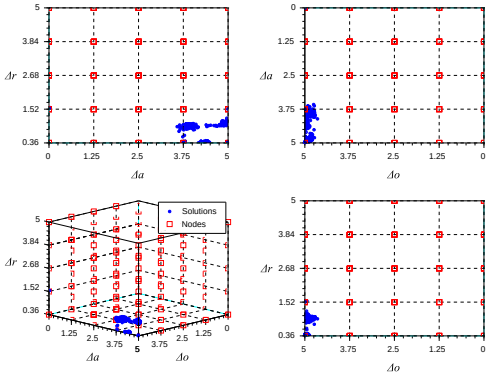
(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space



(e) MOEA/D-DE: Pareto fronts



(f) MOEA/D-DE: Solution space

Figure 4.17: Estimated Pareto fronts and solution spaces in experiment 9

In the Table 4.29 a comparison between multi-objective optimization algorithms after ten independent optimizations using the selected performance metrics is showed.

Table 4.29: Performance of optimization algorithms in experiment 9

Algorithm	MOPSO			NSGA-II-DE			MOEA/D-DE		
Metric	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*
Average	0.7327	0.9173	0.9244	0.7308	0.9100	2.9112	0.7307	0.7895	1.3302
Median	0.7334	0.9227	0.9233	0.7312	0.9182	2.8892	0.7309	0.7864	1.3264
Std. Dev.	0.0016	0.0193	0.0740	0.0020	0.0196	0.0761	0.0017	0.0283	0.0346
Worst	0.7301	0.8909	1.0776	0.7281	0.8682	3.1263	0.7267	0.7409	1.3781
Best	0.7345	0.9364	0.8255	0.7342	0.9273	2.8675	0.7328	0.8409	1.2807

In the Figure 4.18 the relationships between control parameters (Δr, Δo, Δa) and objective functions (f_1, f_2, f_3, f_4) in the experiment of twenty quadrotors without obstacles can be seen.

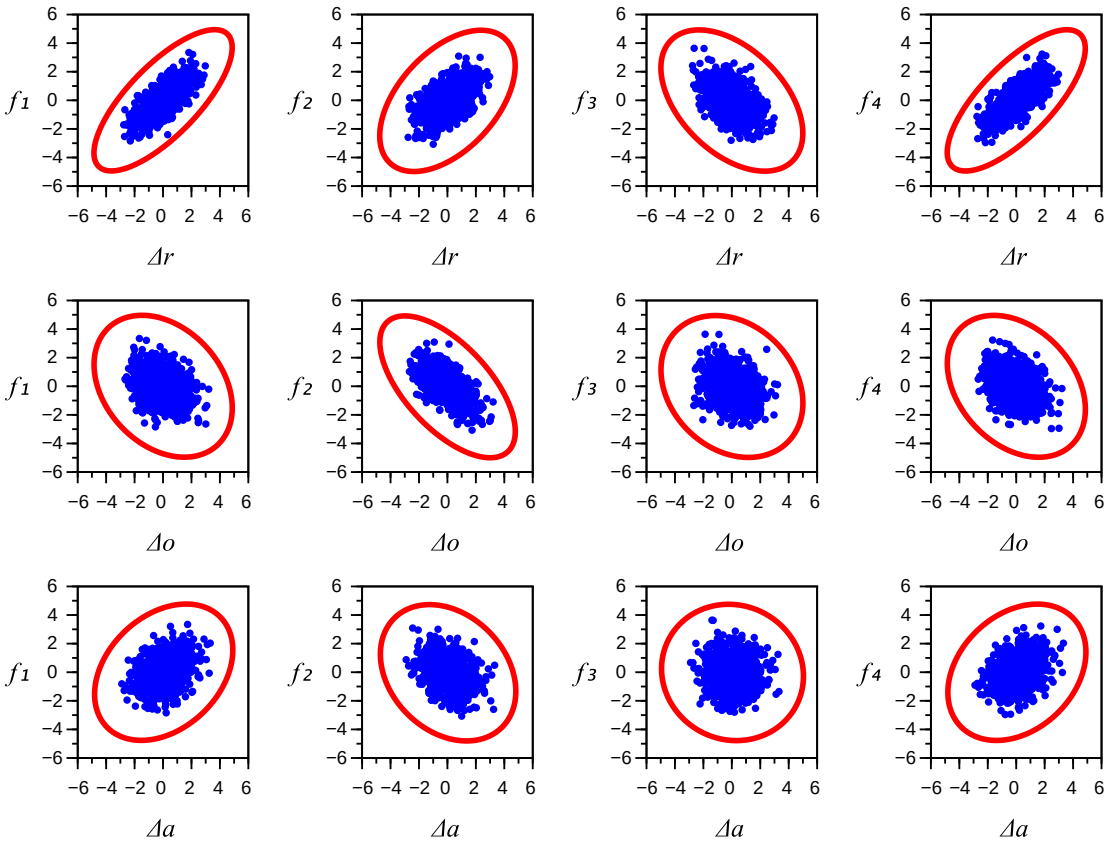


Figure 4.18: Relationship between parameters and functions in experiment 9

Confidence intervals using bootstrap methods are calculated for five random solutions of taken from the best independent optimization in terms of hypervolume measure using MOPSO as showed in the Table 4.30.

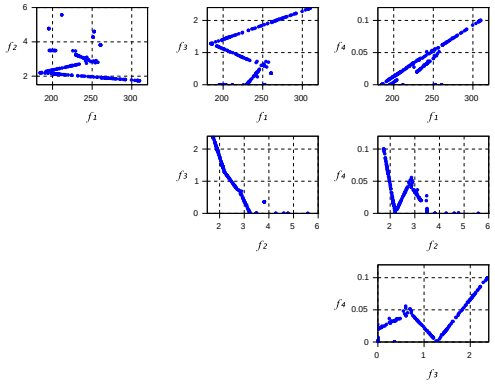
Table 4.30: Confidence intervals in experiment 9

Solution	$MOPSO_5$ ($\Delta_r = 0.3600$, $\Delta_o = 5.0000$, $\Delta_a = 4.3116$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	172.9387	152.3793	152.4406	14.9075	$[121.8934, 182.8651]$
f_2	3.6884	3.9383	3.9357	0.3027	$[3.3192, 4.5575]$
f_3	3.7396	4.2800	4.2718	0.7229	$[2.8017, 5.7583]$
f_4	0.0001	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
Solution	$MOPSO_{10}$ ($\Delta_r = 0.9384$, $\Delta_o = 5.0000$, $\Delta_a = 4.3699$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	190.2324	161.1891	161.0428	11.3345	$[138.0099, 184.3682]$
f_2	5.6788	5.0513	5.0521	0.3556	$[4.3242, 5.7784]$
f_3	2.1952	0.1100	0.1111	0.0618	$[-0.0164, 0.2364]$
f_4	0.0077	0.0083	0.0083	0.0082	$[-0.0084, 0.0251]$
Solution	$MOPSO_{107}$ ($\Delta_r = 0.9885$, $\Delta_o = 4.9999$, $\Delta_a = 4.2933$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	199.8337	173.8905	173.9117	14.8927	$[143.4349, 204.3461]$
f_2	6.1391	5.2898	5.2896	0.3759	$[4.5211, 6.0586]$
f_3	1.2714	0.0800	0.0801	0.0301	$[0.0185, 0.1415]$
f_4	0.0155	0.0083	0.0083	0.0082	$[-0.0084, 0.0251]$
Solution	$MOPSO_{117}$ ($\Delta_r = 0.9470$, $\Delta_o = 5.0000$, $\Delta_a = 4.3089$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	193.5898	198.3878	198.4443	17.5408	$[162.5168, 234.2587]$
f_2	5.6466	5.6448	5.6441	0.3624	$[4.9037, 6.3859]$
f_3	1.8438	0.0633	0.0633	0.0229	$[0.0165, 0.1102]$
f_4	0.0115	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
Solution	$MOPSO_{137}$ ($\Delta_r = 1.0270$, $\Delta_o = 4.9999$, $\Delta_a = 5.0000$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	193.9557	183.2687	183.3462	13.0925	$[156.4945, 210.0429]$
f_2	6.6562	5.2641	5.2639	0.3332	$[4.5827, 5.9455]$
f_3	0.8243	0.0383	0.0380	0.0204	$[-0.0033, 0.0800]$
f_4	0.0001	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$

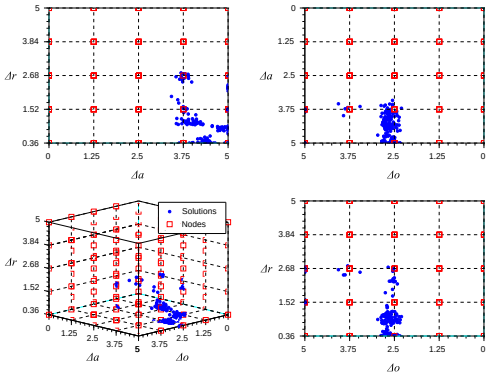
Table 4.30 shows that in experiment 3, 80% of selected solutions has a valid estimation in f_1, 60% of selected solutions has a valid estimation in f_2, 20% of selected solutions has a valid estimation in f_3 and 40% of selected solutions has a valid estimation in f_4.

4.2.4 Experiment 10

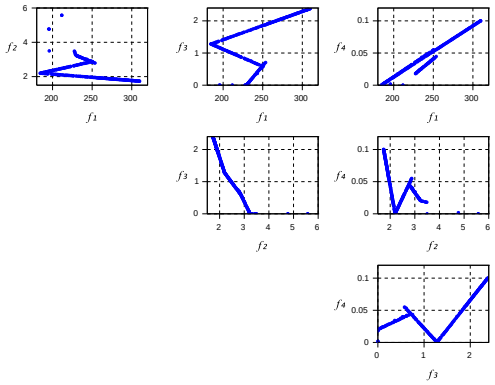
In the Figure 4.19 the estimated Pareto fronts and the solution spaces obtained by each multi-objective optimization algorithm are showed.



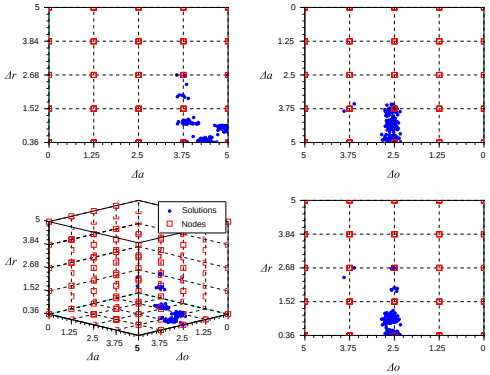
(a) MOPSO: Pareto fronts



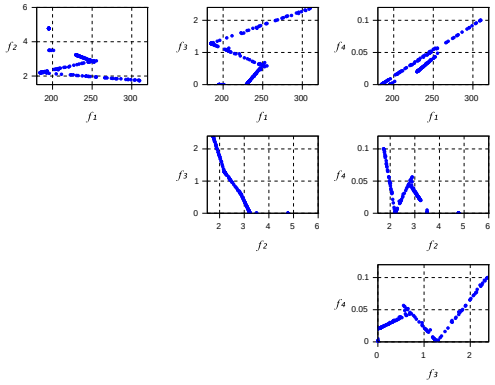
(b) MOPSO: Solution space



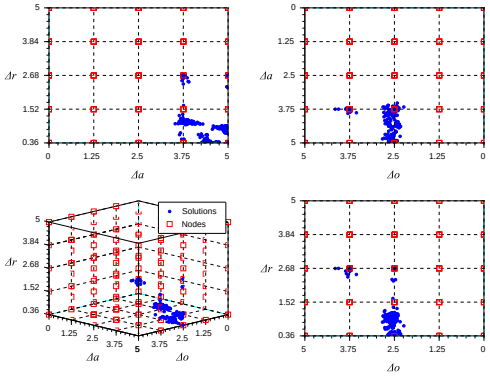
(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space



(e) MOEA/D-DE: Pareto fronts



(f) MOEA/D-DE: Solution space

Figure 4.19: Estimated Pareto fronts and solution spaces in experiment 10

In the Table 4.31 a comparison between multi-objective optimization algorithms after ten independent optimizations using the selected performance metrics is showed.

Table 4.31: Performance of optimization algorithms in experiment 10

Algorithm	MOPSO			NSGA-II-DE			MOEA/D-DE		
Metric	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*
Average	0.7866	0.7182	0.8784	0.7886	0.9264	2.8071	0.7857	0.8291	1.3881
Median	0.7867	0.7205	0.8443	0.7887	0.9273	2.8083	0.7858	0.8318	1.3926
Std. Dev.	0.0005	0.0390	0.0694	0.0002	0.0140	0.0087	0.0005	0.0296	0.0336
Worst	0.7858	0.6591	0.9735	0.7883	0.9091	2.8178	0.7844	0.7864	1.4406
Best	0.7872	0.7773	0.7955	0.7888	0.9545	2.7949	0.7864	0.8727	1.3437

In the Figure 4.20 the relationships between control parameters (Δr, Δo, Δa) and objective functions (f_1, f_2, f_3, f_4) in the experiment of five quadrotors with obstacles can be seen.

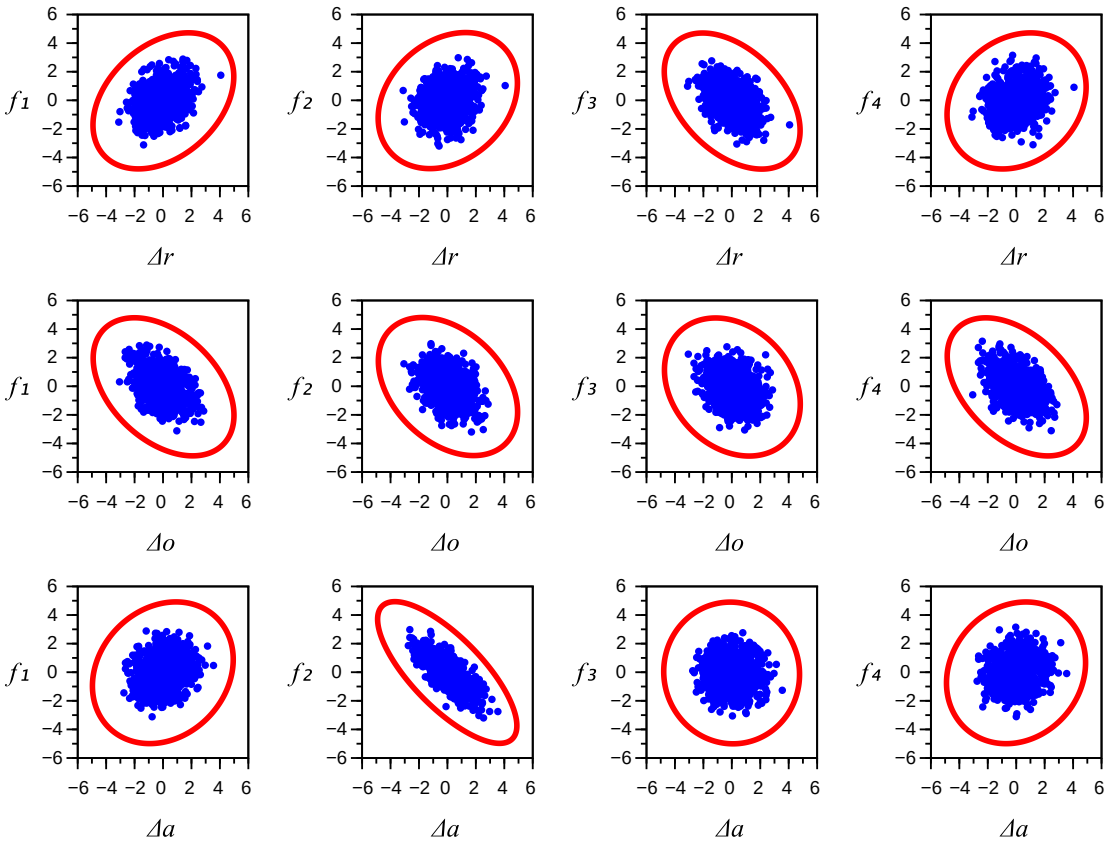


Figure 4.20: Relationship between parameters and functions in experiment 10

Confidence intervals using bootstrap methods are calculated for five random solutions of taken from the best independent optimization in terms of hypervolume measure using NSGA-II-DE as showed in the Table 4.32.

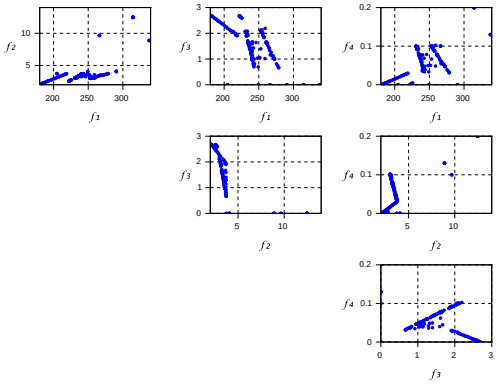
Table 4.32: Confidence intervals in experiment 10

Solution	NSGA-II-DE$_{30}$ ($\Delta_r = 0.3652$, $\Delta_o = 2.6213$, $\Delta_a = 4.4115$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	225.5830	291.8443	291.8170	28.2731	$[234.0259, 349.6627]$
f_2	2.0476	2.5941	2.5923	0.4355	$[1.7035, 3.4846]$
f_3	1.6384	2.1867	2.1835	0.3936	$[1.3818, 2.9915]$
f_4	0.0329	0.1400	0.1393	0.0573	$[0.0229, 0.2571]$
Solution	NSGA-II-DE$_{45}$ ($\Delta_r = 0.3874$, $\Delta_o = 2.5469$, $\Delta_a = 4.3544$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	259.7488	261.8437	262.1668	28.3150	$[203.9394, 319.7479]$
f_2	1.9207	1.8068	1.8066	0.0862	$[1.6304, 1.9831]$
f_3	1.9385	1.5200	1.5227	0.2679	$[0.9722, 2.0678]$
f_4	0.0601	0.0333	0.0330	0.0327	$[-0.0336, 0.1002]$
Solution	NSGA-II-DE$_{79}$ ($\Delta_r = 1.0013$, $\Delta_o = 2.5143$, $\Delta_a = 4.1366$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	251.0874	281.5583	281.5330	27.4991	$[225.3227, 337.7940]$
f_2	2.8431	4.3166	4.3273	0.5628	$[3.1656, 5.4676]$
f_3	0.6250	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0420	0.1067	0.1062	0.0496	$[0.0052, 0.2082]$
Solution	NSGA-II-DE$_{127}$ ($\Delta_r = 1.0112$, $\Delta_o = 2.6496$, $\Delta_a = 3.9845$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	246.9341	276.6980	276.8422	28.6964	$[218.0139, 335.3821]$
f_2	2.9106	3.3862	3.3826	0.4968	$[2.3703, 4.4021]$
f_3	0.5166	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$
f_4	0.0377	0.0600	0.0602	0.0365	$[-0.0147, 0.1347]$
Solution	NSGA-II-DE$_{193}$ ($\Delta_r = 0.3954$, $\Delta_o = 2.6050$, $\Delta_a = 4.4723$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	200.5985	238.1957	238.1052	25.8686	$[185.2944, 291.0970]$
f_2	2.1401	2.6000	2.6049	0.3846	$[1.8135, 3.3866]$
f_3	1.4193	1.4867	1.4933	0.4114	$[0.6454, 2.3280]$
f_4	0.0130	0.0733	0.0729	0.0456	$[-0.0200, 0.1667]$

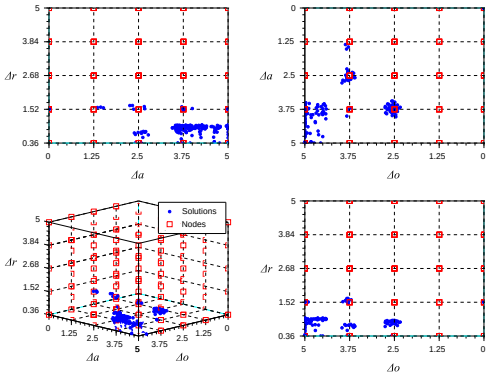
Table 4.32 shows that in experiment 4, 80% of selected solutions has a valid estimation in f_1, 80% of selected solutions has a valid estimation in f_2, 60% of selected solutions has a valid estimation in f_3 and 100% of selected solutions has a valid estimation in f_4.

4.2.5 Experiment 11

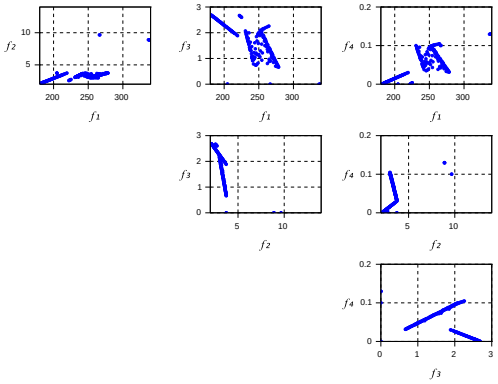
In the Figure 4.21 the estimated Pareto fronts and the solution spaces obtained by each multi-objective optimization algorithm are showed.



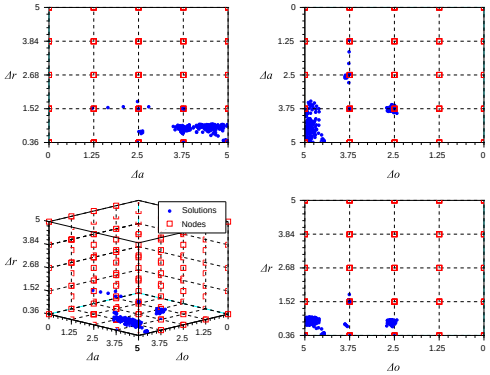
(a) MOPSO: Pareto fronts



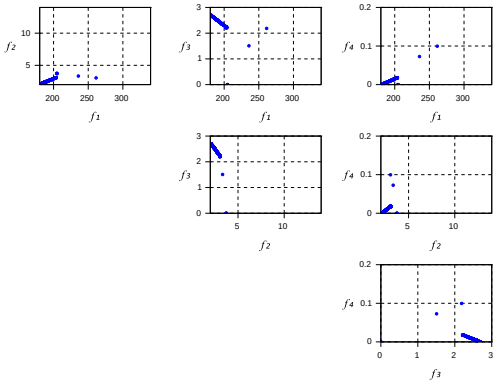
(b) MOPSO: Solution space



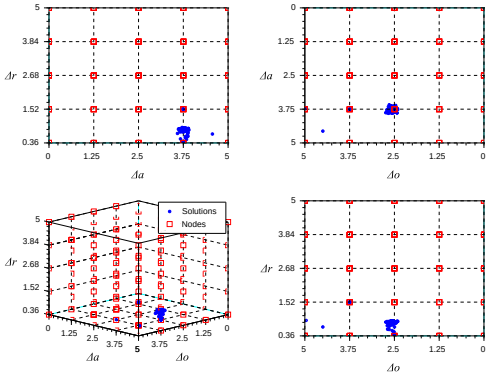
(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space



(e) MOEA/D-DE: Pareto fronts



(f) MOEA/D-DE: Solution space

Figure 4.21: Estimated Pareto fronts and solution spaces in experiment 11

In the Table 4.33 a comparison between multi-objective optimization algorithms after ten independent optimizations using the selected performance metrics is showed.

Table 4.33: Performance of optimization algorithms in experiment 11

Algorithm	MOPSO			NSGA-II-DE			MOEA/D-DE		
Metric	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*
Average	0.7979	0.7445	0.9040	0.8007	0.9068	2.8907	0.7936	0.9305	0.7669
Median	0.7983	0.7500	0.9024	0.8007	0.9045	2.8929	0.7935	0.9500	0.7633
Std. Dev.	0.0009	0.0554	0.0182	0.0000	0.0204	0.0084	0.0012	0.0371	0.0304
Worst	0.7965	0.6591	0.9340	0.8006	0.8818	2.9051	0.7920	0.8636	0.8127
Best	0.7990	0.8500	0.8769	0.8008	0.9500	2.8761	0.7954	0.9636	0.7208

In the Figure 4.22 the relationships between control parameters (Δr, Δo, Δa) and objective functions (f_1, f_2, f_3, f_4) in the experiment of ten quadrotors with obstacles can be seen.

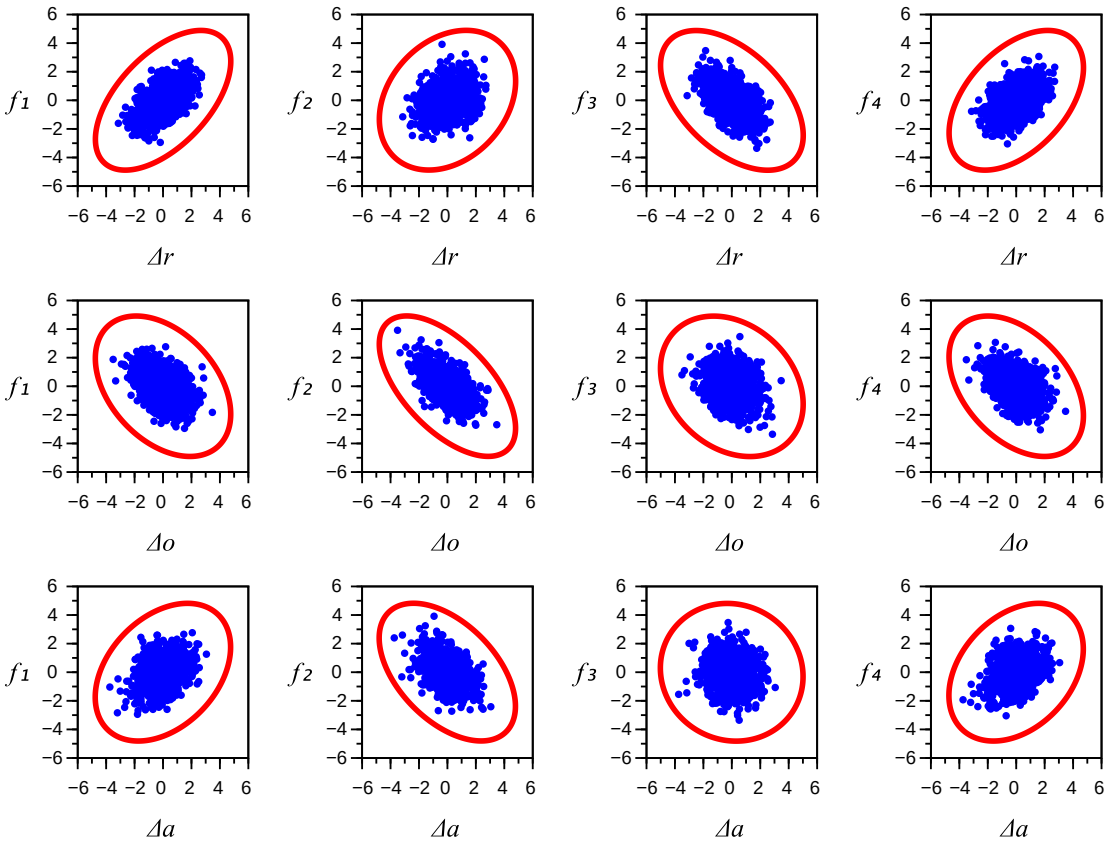


Figure 4.22: Relationship between parameters and functions in experiment 11

Confidence intervals using bootstrap methods are calculated for five random solutions of taken from the best independent optimization in terms of hypervolume measure using NSGA-II-DE as showed in the Table 4.34.

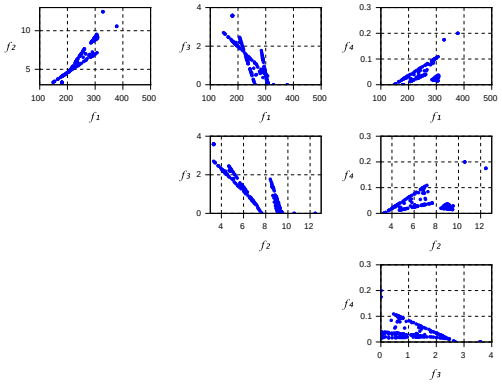
Table 4.34: Confidence intervals in experiment 11

Solution	NSGA-II-DE$_{58}$ ($\Delta_r = 0.9581$, $\Delta_o = 4.8561$, $\Delta_a = 4.9923$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	274.8387	208.9727	208.8247	26.8720	$[154.0195, 263.9258]$
f_2	3.6236	3.6791	3.6782	0.1902	$[3.2901, 4.0681]$
f_3	0.8725	0.0033	0.0033	0.0032	$[-0.0033, 0.0100]$
f_4	0.0413	0.0567	0.0569	0.0397	$[-0.0246, 0.1379]$
Solution	NSGA-II-DE$_{84}$ ($\Delta_r = 0.9608$, $\Delta_o = 4.6810$, $\Delta_a = 4.3476$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	254.4086	242.7853	242.8545	27.4918	$[186.5646, 299.0061]$
f_2	3.6251	3.8881	3.8902	0.2420	$[3.3933, 4.3830]$
f_3	0.9320	0.0033	0.0034	0.0033	$[-0.0034, 0.0101]$
f_4	0.0432	0.0967	0.0970	0.0534	$[-0.0126, 0.2059]$
Solution	NSGA-II-DE$_{151}$ ($\Delta_r = 0.9422$, $\Delta_o = 4.8675$, $\Delta_a = 4.8804$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	271.9405	221.3802	221.5972	21.0471	$[178.3387, 264.4216]$
f_2	3.5427	3.6994	3.7005	0.1437	$[3.4056, 3.9932]$
f_3	1.0333	0.1667	0.1679	0.1163	$[-0.0711, 0.4044]$
f_4	0.0490	0.0033	0.0034	0.0033	$[-0.0034, 0.0100]$
Solution	NSGA-II-DE$_{200}$ ($\Delta_r = 0.7003$, $\Delta_o = 3.8355$, $\Delta_a = 2.6123$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	223.1061	243.5305	243.4696	23.1295	$[196.2307, 290.8303]$
f_2	2.6179	4.5697	4.5637	0.4710	$[3.6064, 5.5329]$
f_3	2.6453	0.2000	0.2003	0.0603	$[0.0768, 0.3233]$
f_4	0.0013	0.0067	0.0066	0.0065	$[-0.0066, 0.0199]$
Solution	NSGA-II-DE$_{215}$ ($\Delta_r = 0.9382$, $\Delta_o = 4.6721$, $\Delta_a = 3.6563$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	239.8864	220.6687	220.6561	16.2074	$[187.5246, 253.8127]$
f_2	3.5603	3.9488	3.9503	0.2564	$[3.4245, 4.4732]$
f_3	1.0745	0.0033	0.0033	0.0033	$[-0.0034, 0.0100]$
f_4	0.0504	0.0000	0.0000	0.0000	$[0.0000, 0.0000]$

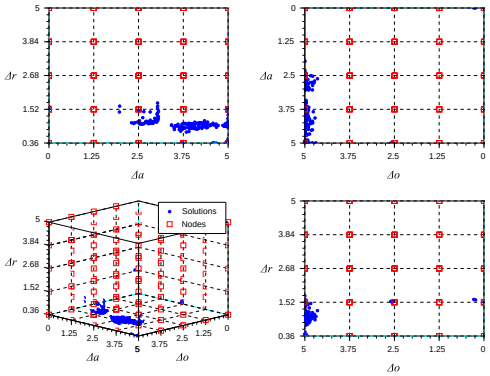
Table 4.34 shows that in experiment 5, 60% of selected solutions has a valid estimation in f_1, 80% of selected solutions has a valid estimation in f_2, 0% of selected solutions has a valid estimation in f_3 and 60% of selected solutions has a valid estimation in f_4.

4.2.6 Experiment 12

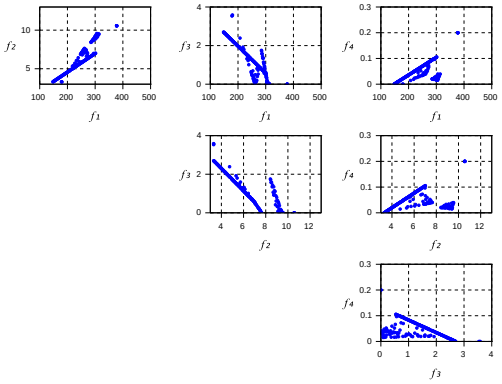
In the Figure 4.23 the estimated Pareto fronts and the solution spaces obtained by each multi-objective optimization algorithm are showed.



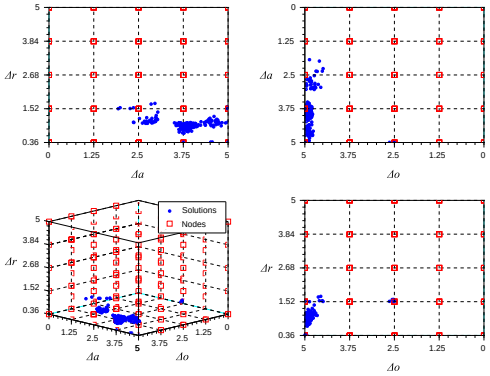
(a) MOPSO: Pareto fronts



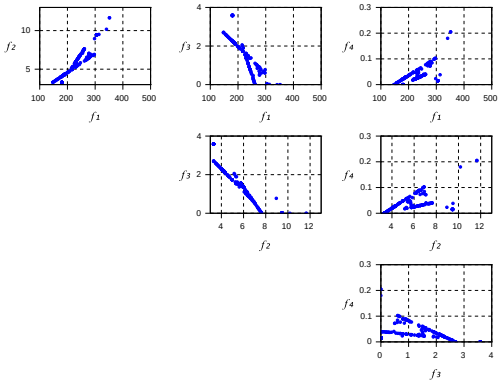
(b) MOPSO: Solution space



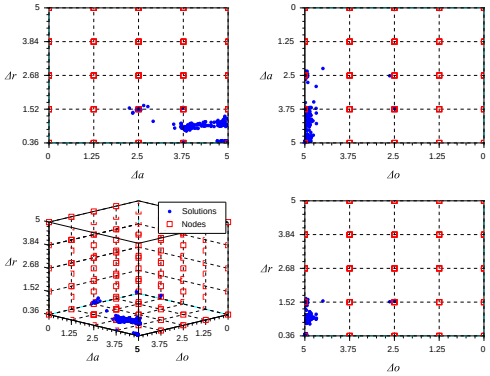
(c) NSGA-II-DE: Pareto fronts



(d) NSGA-II-DE: Solution space



(e) MOEA/D-DE: Pareto fronts



(f) MOEA/D-DE: Solution space

Figure 4.23: Estimated Pareto fronts and solution spaces in experiment 12

In the Table 4.35 a comparison between multi-objective optimization algorithms after ten independent optimizations using the selected performance metrics is showed.

Table 4.35: Performance of optimization algorithms in experiment 12

Algorithm	MOPSO			NSGA-II-DE			MOEA/D-DE		
Metric	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*	*HV*	*Q*	*TPC*
Average	0.7117	0.9141	0.9449	0.7116	0.9000	2.8997	0.7096	0.9091	1.0460
Median	0.7118	0.9159	0.9441	0.7116	0.9023	2.8988	0.7119	0.9000	1.0296
Std. Dev.	0.0013	0.0164	0.0221	0.0023	0.0189	0.0120	0.0054	0.0285	0.0515
Worst	0.7095	0.8864	0.9803	0.7089	0.8727	2.9182	0.6988	0.8727	1.1549
Best	0.7134	0.9364	0.9140	0.7153	0.9364	2.8831	0.7132	0.9591	0.9948

In the Figure 4.24 the relationships between control parameters (Δr, Δo, Δa) and objective functions (f_1, f_2, f_3, f_4) in the experiment of twenty quadrotors with obstacles can be seen.

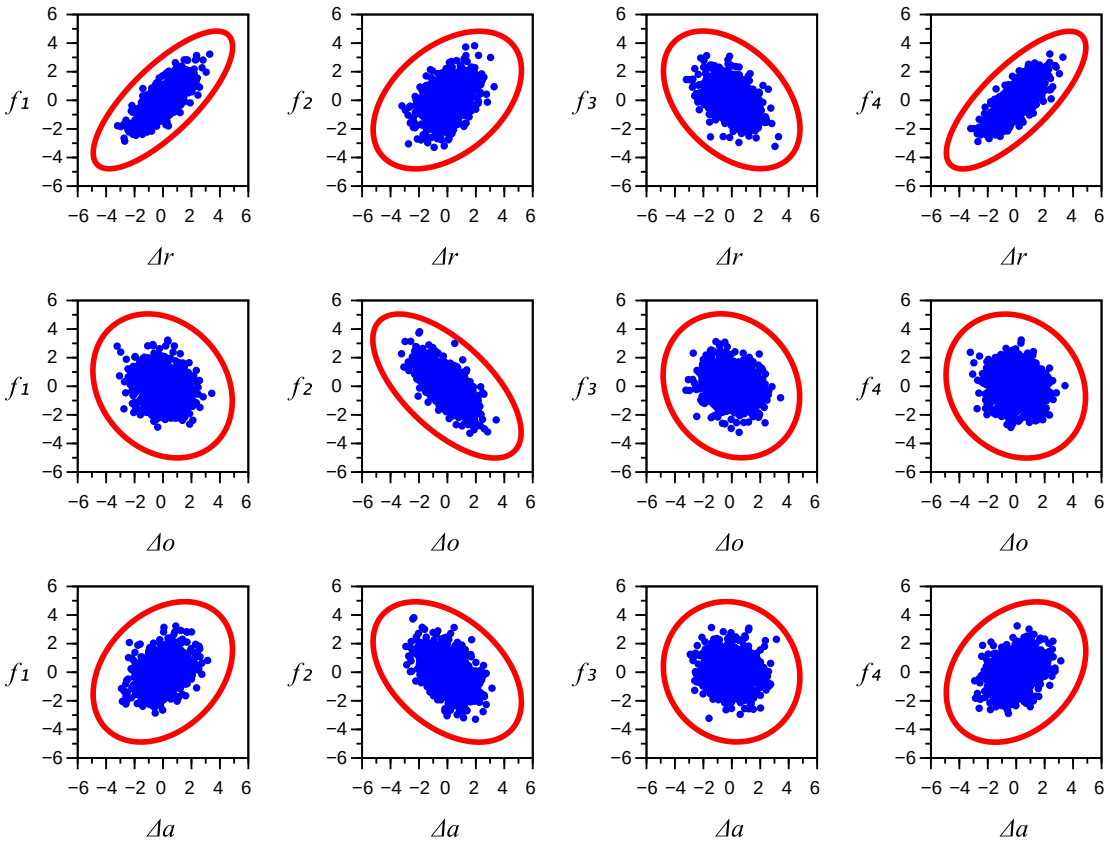


Figure 4.24: Relationship between parameters and functions in experiment 12

Confidence intervals using bootstrap methods are calculated for five random solutions of taken from the best independent optimization in terms of hypervolume measure using MOPSO as showed in the Table 4.36.

Table 4.36: Confidence intervals in experiment 12

Solution	$MOPSO_{10}$ ($\Delta_r = 1.1512$, $\Delta_o = 5.0000$, $\Delta_a = 3.0297$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	302.4008	251.8957	251.8326	19.5868	[211.8407, 291.9508]
f_2	8.8974	6.4752	6.4691	0.4747	[5.5046, 7.4459]
f_3	0.4072	0.0100	0.0099	0.0068	[−0.0038, 0.0238]
f_4	0.0321	0.0500	0.0499	0.0282	[−0.0076, 0.1076]
Solution	$MOPSO_{13}$ ($\Delta_r = 0.8817$, $\Delta_o = 5.0000$, $\Delta_a = 3.7515$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	189.3795	203.1880	203.2280	21.8007	[158.6056, 247.7704]
f_2	4.3064	4.3281	4.3269	0.1656	[3.9894, 4.6669]
f_3	2.1177	0.1350	0.1349	0.0737	[−0.0158, 0.2858]
f_4	0.0285	0.0517	0.0516	0.0368	[−0.0235, 0.1269]
Solution	$MOPSO_{120}$ ($\Delta_r = 1.0142$, $\Delta_o = 5.0000$, $\Delta_a = 5.0000$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	246.2436	230.3197	230.1943	24.5166	[180.1832, 280.4561]
f_2	6.8371	4.9861	4.9858	0.3311	[4.3091, 5.6632]
f_3	0.6632	0.0183	0.0183	0.0080	[0.0020, 0.0346]
f_4	0.0326	0.0750	0.0747	0.0388	[−0.0043, 0.1543]
Solution	$MOPSO_{131}$ ($\Delta_r = 0.8582$, $\Delta_o = 4.9594$, $\Delta_a = 4.0539$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	181.4347	242.3474	242.2870	22.8591	[195.6005, 289.0943]
f_2	4.1143	5.7740	5.7720	0.4820	[4.7882, 6.7597]
f_3	2.2367	0.1950	0.1962	0.0711	[0.0497, 0.3404]
f_4	0.0228	0.0883	0.0881	0.0418	[0.0028, 0.1739]
Solution	$MOPSO_{149}$ ($\Delta_r = 0.9277$, $\Delta_o = 4.8882$, $\Delta_a = 3.6758$)				
Objectives	Estimated	30 Runs	$mean_{boot}$	SE_{boot}	95% Confidence Interval
f_1	224.8379	247.4091	247.3461	22.3086	[201.7879, 293.0302]
f_2	5.1618	5.3388	5.3422	0.3620	[4.5984, 6.0792]
f_3	1.6248	0.1533	0.1542	0.1110	[−0.0736, 0.3803]
f_4	0.0528	0.0067	0.0067	0.0065	[−0.0067, 0.0200]

Table 4.36 shows that in experiment 6, 60% of selected solutions has a valid estimation in f_1, 40% of selected solutions has a valid estimation in f_2, 0% of selected solutions has a valid estimation in f_3 and 80% of selected solutions has a valid estimation in f_4.

Chapter 5

Conclusion

5.1 Discussion

The nondominated solutions found in the experiments 1 to 6 show that the optimized values for Δr are close to the lower bound and the optimized values for Δo and Δa are close to the upper bounds. The comparison between the objective functions obtained by solutions without cooperation between members and the obtained by nondominated solutions demonstrate that the implemented bio-inspired model produces collaborative tendencies among members of the swarm which favor the resolution of the task with a better performance. An important obtained result in the experiments 1 to 6 is the minimum required value for the width of the zone of repulsion that produces zero collisions.

The Pareto fronts showed in the Figures 4.1, 4.3, 4.5, 4.7, 4.9 and 4.11 show a proportional relation between f_1 and f_2 in swarms of many members, this relation is remarked in the experiments 3 and 6, this result suggests that an unite swarm can solve the proposed task in a more efficient way when many members are involved. In most of the cases the total of explorers are minimum when the others three functions are minimized because when all the members reach the zone of influence a good evaluation in all the objective functions are expected.

Tables 4.3, 4.7, 4.11, 4.15, 4.19 and 4.23 show that in the experiments 1, 2 and 3 the MOPSO produces more nondominated solutions than the NSGA-II-DE and in the experiments 4, 5 and 6 the opposite occurs, however in the experients 1, 4, 5 and 6 the MOPSO obtains a better evaluation in HV and Q than NSGA-II-DE and in experiments 2 and 3 the NSGA-II-DE obtains a better evaluation in HV and Q than MOPSO and the MOPSO obtains a better evaluation in terms of TPC in the experiments 1 to 6; all previous results suggest that in a real life multi-objective problem the MOPSO algorithm outperforms the NSGA-II-DE algorithm. However, this can be due to the lack of population and cycles which have a more negative effect in genetic algorithms than collective intelligence algorithms, this relationship between algorithms performance and computational cost is also seen in single-objective optimization [74].

Confidence intervals from Tables 4.4, 4.8, 4.12, 4.16, 4.20 and 4.24 confirm that 10 runs per solution give a good approximation of the true objective functions in the experiments 1 to 6.

Figures 4.13, 4.15, 4.17, 4.19, 4.21 and 4.23 show that in the experiments 7 to 12 most of the found solutions are contained in smaller ranges than the original proposed search space, for the radius of the zone of repulsion most solutions are in the following range $\Delta r = [0.36, 2.68]$, for the zone of orientation are in the following range $\Delta o = [2.5, 5]$ and for the zone of attraction are in the following range $\Delta a = [2.5, 5]$; some found solutions are outside of the previous ranges but the number of solutions with this condition are minimum. The results from the solution spaces suggest that the original search space $[0.36, 5] \times [0, 5]^2$ is too large because the most of optimal solutions are in a small region of the entire search space, this denotes that small values for the parameter Δr are preferred and large values for the parameters Δo and Δa are preferred in a flocking taskwith target zone search which is in accordance with the results obtained in experiments 1 to 6.

Estimated Pareto fronts show that the objectives f_2 and f_3 are in conflict in the experiments 7 to 12, the objectives f_3 and f_4 are in conflict when are obstacles in the arena and the swarm has ten or twenty members, the objectives f_1 and f_3 are in conflict mainly with swarms of ten or twenty members regardless the presence of obstacles, the objectives f_1 and f_2 are in conflict mainly when do not exists obstacles in the arena and it is more evident with small swarms, the objectives f_2 and f_4 are in conflict when the swarm is small and exists obstacles in the arena, the objectives f_1 and f_4 do not show conflict between them.

As can seen in the Tables 4.25, 4.27, 4.29, 4.31, 4.33 and 4.35 the following general results are obtained. In terms of HV the MOPSO was the best evaluated in the experiments 8, 9 and 12; the NSGA-II-DE was the best evaluated in experiments 7, 10 and 11 but was the worst evaluated in the experiment 8; the MOEA/D-DE was the worst evaluated in the experiments 7, 9, 10, 11 and 12. In terms of Q the MOPSO was the best evaluated in the experiments 9 and 12 but was the worst evaluated in the experiments 7, 10 and 11; the NSGA-II-DE was the best evaluated in experiments 7 and 10 but was the worst evaluated in the experiments 8 and 12; the MOEA/D-DE was the best evaluated in experiments 8 and 11 but was the worst evaluated in the experiment 9. In terms of TPC the MOPSO was the best evaluated in the experiments 7, 8, 9, 10 and 12; the NSGA-II-DE was the worst evaluated in the experiments 7 to 12; the MOEA/D-DE was the best evaluated in experiment 11. With these results is clearly that in the experiment 7 to 12 the MOPSO is the best algorithm in the HV metric but is the worst algorithm in terms of Q metric. With MOEA/D is the opposite case, it is the best algorithm in Q metric but is the worst evaluated algorithm in HV metric. The algorithm NSGA-II-DE is the most balanced algorithm in terms of HV and Q metrics. About the TPC metric, the MOPSO is the best evaluated and NSGA-II-DE is the worst evaluated algorithm.

As showed in the Figures 4.14, 4.16, 4.18, 4.20, 4.22 and 4.24 the following relations between control parameters and objectives are obtained. An increase in Δr is related to a decrease in f_3 but is related to an increase in f_1, f_2 and f_4, these relations are more significant as the number of members in the swarm increase. An increase in Δo is related to a decrease in all the objectives, this relation with the objective f_3 is not very significant and in the swarm of 10 members with obstacles this relation with the objective f_4 is not very significant. An increase in Δa is related to a decrease in f_2, this relation is more significant with small number of members in the swarm but in the experiments with obstacles is observed that an increase in Δa is related to a increase in f_1 and f_4. These results show that the zone of repulsion is only needed in order to reduce the number of collisions between members but this zone harms the minimization of the other objectives, the zone of orientation is very important in the location-aggregation task because it helps to minimize all the objectives, the zone of attraction is important too, but it harms slightly the minimization of the objective f_1 and f_4.

As can seen in the Tables 4.26, 4.28, 4.30, 4.32, 4.34 and 4.36 the objective function f_1 has 80% or more valid solutions in experiments 7, 8, 9 and 10, f_2 has 80% or more valid solutions in experiments 7, 8, 10 and 11, f_3 has 80% or more valid solutions in none experiment, f_4 has 80% or more valid solutions in experiments 7, 10 and 12. It is necessary add more nodes in the parameter Δr in order to improve the estimation of f_3.

5.2 General conclusions

The flocking task with target zone search was successfully completed by swarms of quadrotors with different number of members and with and without the presence of obstacles in the environment, the sets of optimized control parameters were found in each experiment.

The proposed surrogate model called nodes-based statistical model helps to evaluate the performance of the swarm of quadrotors without the need to evaluate each candidate solution directly in the simulator, this allows the use of multi-objective optimization algorithms with large populations and large number of cycles or generations which produces estimated Pareto fronts and solution spaces with enough information to observe the relationship between objectives and the optimized control parameters which solve the proposed task with a very good performance.

The small differences between estimated solution spaces of the different proposed experiments highlights the scalability and robustness of the model but the importance of finding the correct sets of control parameters in order to obtain the best performance in each case is more obvious when observing the differences in the estimated Pareto fronts. The impact of the control parameters of the swarm on the proposed objectives in each experiment is showed in order to support the found solution spaces, this helps to determine control parameters of the swarm in order to reach a particular behavior taking into consideration the size of the swarm and the conditions of the environment. The nodes-based statistical model shows to be a good option to estimate the objective functions of tasks which simulation requires high computational time, for that reason this model can be implemented in the optimization of more tasks developed by swarms of quadrotors.

The comparison between multi-objective optimization algorithms in the proposed experiments shows that the MOPSO is the best option for optimize real life multi-objective problems when limited population and cycles are used, however, when there are not limitations about computational time during the optimization process the NSGA-II-DE is the most convenient algorithm due to the easy parametrization and the good performance in HV and Q metrics.

The good performance of the bio-inspired model for steer a swarm of quadrotors in a flocking task highlights the possibility of use this model in a wide variety of real-life problems as parcel delivery or construction tasks.

5.3 Future work

The optimization of the task with more nodes in the parameter Δr must be done as future work to improve the estimation of the number of collisions. A detailed methodology for select the number of nodes per dimension must be proposed and a dynamic τ parameter which depends on the position of the solution with respect to the neighbors must be added to the nodes-based statistical model in order to improve the estimation of objective functions, larger values of τ are preferred when the solution is in the middle of neighbors and smaller values of τ are preferred when the solution is near of a neighbor. A comparison between the nodes-based statistical model and others techniques to estimate the objective functions is needed in order to determine the true scope of the proposed surrogate model.

More tasks for swarms, as predator-prey or construction tasks, must be optimized in order to verify the differences between optimized control parameters in different types of tasks.

A perception system that allows the bio-inspired model to be fully applied must be incorporated into real quadrotors in order to replicate in real swarms of quadrotors the obtained results. The perception system must be able to recognize neighbors and obstacles, perceive influences and detect the orientation of adjacent neighbors.

Appendix A

Optimization algorithms

A.1 Function to use MOPSO

The MOPSO algorithm was implemented in the software Scilab 6.0.1. The Algorithm A.1 explains in a general way the main sections of the implemented function called MOPSO().

Algorithm A.1 Function to use MOPSO

```
1: function [PRep, FERep]=MOPSO(TI, TR, TIRep, TCycles, LI, LS)
2:     Parametrization;
3:     Initialization;
4:     Generate population;
5:     Evaluation;
6:     Update repository;
7:     Update adaptive grid;
8:     for cycle = 1 : TCycles do
9:         for particle = 1 : TI do
10:            Select leader;
11:            Calculate velocity;
12:            Update position;
13:            Mutation;
14:            Evaluation;
15:            Update best;
16:        end for
17:        Update repository;
18:        Update adaptive grid;
19:    end for
20: end function
```

Where TI is the total of particles, TR is the dimension of each particle, $TIRep$ is the maximum size of the repository, $TCycles$ is the total of cycles, LI and LS contain the lower and upper limits of each dimension respectively. The outputs $PRep$ and $FERep$ contain the optimized solutions in the repository and their obtained evaluations respectively.

A.2 Function to use NSGA-II-DE

The NSGA-II-DE was implemented in the software Scilab 6.0.1. The Algorithm A.2 explains in a general way the main sections of the implemented function called NSGAIIDE().

Algorithm A.2 Function to use NSGA-II-DE

```
1: function [PRep, FERep]=NSGAIIDE(TI, TR, TIRep, TGen, LI, LS)
2:     Parametrization;
3:     Initialization;
4:     Generate population;
5:     Evaluation;
6:     Sorting;
7:     Update repository;
8:     for generation = 1 : TGen do
9:         for individual = 1 : TI do
10:            Selection;
11:            Crossover;
12:            Mutation;
13:            Evaluation;
14:        end for
15:        Sorting
16:        Update repository;
17:    end for
18: end function
```

Where $TGen$ is the total of generations and the remaining inputs and outputs means the same as the function MOPSO() with the difference that the solutions are called individuals.

A.3 Function to use MOEA/D-DE

The MOEA/D-DE was implemented in the software Scilab 6.0.1. The Algorithm A.3 explains in a general way the main sections of the implemented function called MOEADDE().

Algorithm A.3 Function to use MOEA/D-DE

```
1: function [PRep, FERep]=MOEADDE(H, TR, TIRep, TGen, LI, LS)
2:     Parametrization;
3:     Initialization;
4:     Generate subproblems;
5:     Generate population;
6:     Evaluation;
7:     Decomposition;
8:     Update repository;
9:     for generation = 1 : TGen do
10:        for individual = 1 : TI do
11:            Selection;
12:            Crossover;
13:            Mutation;
14:            Evaluation;
15:            Decomposition;
16:            Replace individuals;
17:        end for
18:        Update repository;
19:        Update adaptive grid;
20:    end for
21: end function
```

Where H is a integer to select the total of individuals and the remaining inputs and outputs means the same as the function NSGAIIDE().

Appendix B

Results of nodes evaluation

B.1 Nodes for experiment 7

In the Table B.1 the evaluation obtained by each generated node after 10 repetitions in the experiment 7 is showed. The evaluation of all the nodes takes 0.6644 days of computational time.

Table B.1: Nodes evaluation in experiment 7

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
1	0.36	0.00	0.00	248.2920	11.4633	16.4600	0.1600
2	0.36	0.00	1.25	298.3480	12.0115	18.0800	0.1200
3	0.36	0.00	2.50	278.4800	10.8163	19.1400	0.0600
4	0.36	0.00	3.75	334.8950	10.1086	26.5200	0.0600
5	0.36	0.00	5.00	284.0430	8.5678	22.3400	0.0600
6	0.36	1.25	0.00	259.4640	11.7081	11.9800	0.1000
7	0.36	1.25	1.25	179.7590	9.8208	8.7800	0.0200
8	0.36	1.25	2.50	212.5960	4.9576	4.0600	0.1000
9	0.36	1.25	3.75	188.4840	4.7413	4.5800	0.0200
10	0.36	1.25	5.00	335.5310	4.9697	3.8200	0.2400
11	0.36	2.50	0.00	209.8940	10.4740	10.4200	0.0800
12	0.36	2.50	1.25	178.4640	6.7400	6.6600	0.0200
13	0.36	2.50	2.50	204.3700	5.0176	3.1800	0.0000
14	0.36	2.50	3.75	203.1720	3.5346	3.0800	0.0000
15	0.36	2.50	5.00	235.6130	3.0858	3.5200	0.0800

Table B.1: Nodes evaluation in experiment 7 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
16	0.36	3.75	0.00	284.7020	11.0673	8.9000	0.1400
17	0.36	3.75	1.25	191.9510	9.0744	4.8000	0.0200
18	0.36	3.75	2.50	166.8650	4.8285	1.6200	0.0400
19	0.36	3.75	3.75	164.2380	2.4303	0.7400	0.0000
20	0.36	3.75	5.00	166.8060	2.4739	0.9200	0.0600
21	0.36	5.00	0.00	231.1790	10.7088	8.6600	0.0800
22	0.36	5.00	1.25	203.3970	8.5708	3.4000	0.0000
23	0.36	5.00	2.50	204.4930	5.1338	2.5800	0.0200
24	0.36	5.00	3.75	139.8290	3.4695	0.6200	0.0000
25	0.36	5.00	5.00	207.8380	3.0272	0.8800	0.1000
26	1.52	0.00	0.00	289.9390	12.0956	0.0000	0.1400
27	1.52	0.00	1.25	245.3570	10.7649	0.0000	0.0200
28	1.52	0.00	2.50	257.0860	9.1076	0.0000	0.0400
29	1.52	0.00	3.75	251.6490	8.5482	0.0000	0.1000
30	1.52	0.00	5.00	285.7210	10.2467	0.0000	0.1400
31	1.52	1.25	0.00	212.3980	9.9573	0.0000	0.0800
32	1.52	1.25	1.25	218.6140	9.9407	0.0000	0.0200
33	1.52	1.25	2.50	218.4450	6.8326	0.0000	0.1000
34	1.52	1.25	3.75	246.3250	5.7451	0.0000	0.0200
35	1.52	1.25	5.00	214.8150	3.6507	0.0000	0.0600
36	1.52	2.50	0.00	238.0000	11.1554	0.0000	0.0600
37	1.52	2.50	1.25	234.2860	10.5339	0.0000	0.0200
38	1.52	2.50	2.50	141.9130	4.9153	0.0000	0.0400
39	1.52	2.50	3.75	247.5720	4.1323	0.0000	0.0600
40	1.52	2.50	5.00	180.2110	3.2933	0.0000	0.0000
41	1.52	3.75	0.00	254.5230	11.4132	0.0000	0.0600
42	1.52	3.75	1.25	207.0680	7.4442	0.0000	0.1000
43	1.52	3.75	2.50	229.2790	5.5785	0.0000	0.1200
44	1.52	3.75	3.75	143.4800	3.7847	0.0000	0.0000
45	1.52	3.75	5.00	244.3460	3.9968	0.0000	0.0800
46	1.52	5.00	0.00	252.1460	11.0552	0.0000	0.0800
47	1.52	5.00	1.25	251.6510	8.1987	0.0200	0.0400
48	1.52	5.00	2.50	203.0870	4.8107	0.0000	0.0000
49	1.52	5.00	3.75	158.4450	4.1092	0.0000	0.0000
50	1.52	5.00	5.00	206.3290	4.0955	0.0000	0.0000
51	2.68	0.00	0.00	282.7650	11.7410	0.0000	0.1000
52	2.68	0.00	1.25	284.2900	11.0389	0.0000	0.1400
53	2.68	0.00	2.50	322.4800	11.1231	0.0000	0.1000
54	2.68	0.00	3.75	373.7130	9.4071	0.0000	0.2800
55	2.68	0.00	5.00	347.3690	10.2128	0.0000	0.1800

Table B.1: Nodes evaluation in experiment 7 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
56	2.68	1.25	0.00	263.4620	10.8053	0.0000	0.1200
57	2.68	1.25	1.25	273.8300	9.6945	0.0000	0.0400
58	2.68	1.25	2.50	239.8670	8.8114	0.0000	0.0400
59	2.68	1.25	3.75	227.0370	6.2388	0.0000	0.0200
60	2.68	1.25	5.00	193.8420	4.8595	0.0000	0.0600
61	2.68	2.50	0.00	279.7700	11.1636	0.0000	0.1200
62	2.68	2.50	1.25	233.2200	9.9510	0.0000	0.0600
63	2.68	2.50	2.50	317.1220	8.2936	0.0000	0.1200
64	2.68	2.50	3.75	183.6050	5.7056	0.0000	0.0000
65	2.68	2.50	5.00	279.9220	4.6852	0.0000	0.0600
66	2.68	3.75	0.00	197.8600	9.6859	0.0000	0.0400
67	2.68	3.75	1.25	266.3260	9.9003	0.0000	0.1800
68	2.68	3.75	2.50	147.9380	5.5404	0.0000	0.0400
69	2.68	3.75	3.75	205.4050	5.0640	0.0000	0.0000
70	2.68	3.75	5.00	142.6890	4.3905	0.0000	0.0000
71	2.68	5.00	0.00	241.9660	10.4978	0.0000	0.1200
72	2.68	5.00	1.25	195.1700	7.0008	0.0000	0.0000
73	2.68	5.00	2.50	209.1550	6.0542	0.0000	0.0000
74	2.68	5.00	3.75	189.2570	5.1454	0.0000	0.0200
75	2.68	5.00	5.00	131.8460	4.3912	0.0000	0.0000
76	3.84	0.00	0.00	267.8990	11.9337	0.0000	0.1200
77	3.84	0.00	1.25	321.6400	11.3492	0.3600	0.1600
78	3.84	0.00	2.50	394.3960	11.2983	0.5400	0.2400
79	3.84	0.00	3.75	329.0630	10.8473	0.0000	0.1000
80	3.84	0.00	5.00	325.8380	10.6579	0.3800	0.0800
81	3.84	1.25	0.00	279.4430	11.3246	0.0000	0.0400
82	3.84	1.25	1.25	293.7680	11.1802	0.0000	0.1200
83	3.84	1.25	2.50	223.5040	7.8394	0.0000	0.0200
84	3.84	1.25	3.75	268.5740	8.3153	0.2800	0.0800
85	3.84	1.25	5.00	199.3870	4.2634	0.3000	0.0200
86	3.84	2.50	0.00	227.3490	10.4621	0.0000	0.0800
87	3.84	2.50	1.25	218.3180	9.3418	0.0000	0.0800
88	3.84	2.50	2.50	282.4750	9.0449	0.0000	0.0600
89	3.84	2.50	3.75	284.3160	7.5575	0.0000	0.0200
90	3.84	2.50	5.00	170.4820	5.3757	0.0000	0.0200
91	3.84	3.75	0.00	240.6330	10.5098	0.0000	0.1000
92	3.84	3.75	1.25	252.8920	9.5449	0.0000	0.0800
93	3.84	3.75	2.50	200.6360	7.8418	0.0200	0.0200
94	3.84	3.75	3.75	199.7730	6.7393	0.0000	0.0200
95	3.84	3.75	5.00	300.3790	4.7281	0.1800	0.0400

Table B.1: Nodes evaluation in experiment 7 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
96	3.84	5.00	0.00	235.8590	11.9193	0.0000	0.0800
97	3.84	5.00	1.25	236.7450	9.0253	0.0000	0.0800
98	3.84	5.00	2.50	166.0630	6.7682	0.0000	0.0000
99	3.84	5.00	3.75	149.2220	5.8725	0.0000	0.0000
100	3.84	5.00	5.00	182.2170	6.1692	0.0000	0.0000
101	5.00	0.00	0.00	261.6960	11.9074	0.0000	0.1400
102	5.00	0.00	1.25	383.3630	11.6783	0.3600	0.2400
103	5.00	0.00	2.50	354.3180	10.9227	0.3600	0.2400
104	5.00	0.00	3.75	366.5170	10.4586	0.0000	0.2600
105	5.00	0.00	5.00	356.3110	8.3707	0.0800	0.1600
106	5.00	1.25	0.00	228.4170	10.8120	0.0000	0.0600
107	5.00	1.25	1.25	254.0150	10.3336	0.0000	0.0600
108	5.00	1.25	2.50	293.8650	9.3685	0.0000	0.0400
109	5.00	1.25	3.75	290.7280	9.3241	0.2000	0.1000
110	5.00	1.25	5.00	277.1940	7.5548	0.0000	0.0400
111	5.00	2.50	0.00	249.5210	11.4066	0.0000	0.0600
112	5.00	2.50	1.25	267.7970	9.5260	0.3200	0.0600
113	5.00	2.50	2.50	250.2010	8.3830	0.0000	0.0400
114	5.00	2.50	3.75	224.6220	7.9683	0.0000	0.0800
115	5.00	2.50	5.00	236.9850	7.0696	0.0000	0.1000
116	5.00	3.75	0.00	167.4740	9.3579	0.0000	0.0000
117	5.00	3.75	1.25	191.9340	9.1520	0.0000	0.0400
118	5.00	3.75	2.50	226.8530	8.8501	0.0000	0.0200
119	5.00	3.75	3.75	236.0070	7.1807	0.0000	0.0000
120	5.00	3.75	5.00	220.3950	5.5817	0.0000	0.0000
121	5.00	5.00	0.00	225.9090	10.9508	0.0000	0.0200
122	5.00	5.00	1.25	254.0710	9.7313	0.0000	0.1000
123	5.00	5.00	2.50	189.2970	7.3758	0.1800	0.0200
124	5.00	5.00	3.75	251.6280	6.8656	0.0000	0.0600
125	5.00	5.00	5.00	297.0310	6.5362	0.1200	0.1000

B.2 Nodes for experiment 8

In the Table B.2 the evaluation obtained by each generated node after 10 repetitions in the experiment 8 is showed. The evaluation of all the nodes takes 2.0170 days of computational time.

Table B.2: Nodes evaluation in experiment 8

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
1	0.36	0.00	0.00	256.4945	11.3730	36.9600	0.0700
2	0.36	0.00	1.25	270.7550	11.1005	46.2000	0.1500
3	0.36	0.00	2.50	318.2735	11.2118	48.7200	0.1500
4	0.36	0.00	3.75	290.6205	10.3744	46.5800	0.1700
5	0.36	0.00	5.00	398.0515	10.7809	49.1900	0.2900
6	0.36	1.25	0.00	241.4790	10.4713	26.3300	0.0500
7	0.36	1.25	1.25	285.1755	11.0944	21.7200	0.1300
8	0.36	1.25	2.50	231.7035	8.0914	26.8800	0.0400
9	0.36	1.25	3.75	293.3375	9.3363	15.8800	0.2100
10	0.36	1.25	5.00	199.8745	6.5851	17.5300	0.0300
11	0.36	2.50	0.00	248.2610	10.5165	26.1500	0.0600
12	0.36	2.50	1.25	263.2515	8.2363	14.5800	0.0800
13	0.36	2.50	2.50	234.9945	5.6804	13.6700	0.1000
14	0.36	2.50	3.75	225.7440	3.2097	5.9400	0.0100
15	0.36	2.50	5.00	202.0550	5.4326	4.0700	0.0000
16	0.36	3.75	0.00	225.0915	10.1273	25.5300	0.0700
17	0.36	3.75	1.25	232.4135	7.7183	13.2800	0.0400
18	0.36	3.75	2.50	188.3640	3.6481	8.3600	0.0500
19	0.36	3.75	3.75	243.1775	3.9848	5.3000	0.0600
20	0.36	3.75	5.00	174.7235	2.6016	1.6200	0.0000
21	0.36	5.00	0.00	234.9280	10.3467	21.3200	0.0800
22	0.36	5.00	1.25	186.4575	7.5469	12.6100	0.0500
23	0.36	5.00	2.50	210.9650	4.5164	3.9100	0.0200
24	0.36	5.00	3.75	128.3780	3.7391	2.7700	0.0000
25	0.36	5.00	5.00	181.0760	3.7155	3.0000	0.0500
26	1.52	0.00	0.00	274.6970	11.2319	0.0000	0.1000
27	1.52	0.00	1.25	259.4940	10.5960	0.0000	0.0700
28	1.52	0.00	2.50	321.6990	11.5145	0.0000	0.1700
29	1.52	0.00	3.75	326.5125	10.2579	0.0000	0.1900
30	1.52	0.00	5.00	353.9810	11.2531	0.0000	0.2800

Table B.2: Nodes evaluation in experiment 8 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
31	1.52	1.25	0.00	252.3235	10.6909	0.0000	0.0700
32	1.52	1.25	1.25	277.2900	10.5702	0.0000	0.0700
33	1.52	1.25	2.50	293.1410	8.3424	0.0000	0.0600
34	1.52	1.25	3.75	330.1635	9.4826	0.0000	0.0600
35	1.52	1.25	5.00	274.6680	7.7933	0.0000	0.0800
36	1.52	2.50	0.00	234.5940	9.8283	0.0000	0.0700
37	1.52	2.50	1.25	246.7845	9.1307	0.0000	0.0600
38	1.52	2.50	2.50	229.8145	8.1827	0.0000	0.0000
39	1.52	2.50	3.75	265.7490	8.0468	0.0000	0.0200
40	1.52	2.50	5.00	205.4780	6.0144	0.0000	0.0100
41	1.52	3.75	0.00	234.1100	10.6091	0.0000	0.0600
42	1.52	3.75	1.25	241.1415	9.0624	0.0000	0.0800
43	1.52	3.75	2.50	226.6640	7.4993	0.0000	0.1100
44	1.52	3.75	3.75	172.9735	4.5695	0.0000	0.0000
45	1.52	3.75	5.00	179.2115	3.8754	0.0000	0.0000
46	1.52	5.00	0.00	251.0550	10.7874	0.0000	0.0600
47	1.52	5.00	1.25	229.1170	7.8204	0.0000	0.0400
48	1.52	5.00	2.50	237.9950	8.3793	0.0000	0.0400
49	1.52	5.00	3.75	178.3180	4.9671	0.0000	0.0200
50	1.52	5.00	5.00	167.6310	4.2231	0.0000	0.1000
51	2.68	0.00	0.00	265.1375	11.2625	0.0000	0.1000
52	2.68	0.00	1.25	348.9270	12.1986	0.0500	0.2200
53	2.68	0.00	2.50	364.0045	12.0375	0.0000	0.2200
54	2.68	0.00	3.75	396.6655	12.2399	0.0000	0.3100
55	2.68	0.00	5.00	301.5950	10.8033	0.2100	0.1400
56	2.68	1.25	0.00	265.2455	10.8801	0.0000	0.1000
57	2.68	1.25	1.25	318.8935	11.0800	0.0000	0.1700
58	2.68	1.25	2.50	294.3120	9.6095	0.0200	0.0900
59	2.68	1.25	3.75	368.2940	9.7065	0.0200	0.2100
60	2.68	1.25	5.00	349.6330	9.3189	0.0700	0.2000
61	2.68	2.50	0.00	281.6560	11.3821	0.0000	0.1200
62	2.68	2.50	1.25	318.8110	10.1626	0.3700	0.1300
63	2.68	2.50	2.50	322.7275	9.9033	0.1200	0.1500
64	2.68	2.50	3.75	295.3990	6.9913	0.3200	0.0400
65	2.68	2.50	5.00	290.6970	8.0221	0.2200	0.0700
66	2.68	3.75	0.00	258.4360	10.6006	0.0000	0.1200
67	2.68	3.75	1.25	250.7725	9.3788	0.0000	0.0300
68	2.68	3.75	2.50	273.7085	8.3369	0.1100	0.0600
69	2.68	3.75	3.75	277.9905	6.9228	0.0600	0.0000
70	2.68	3.75	5.00	171.4790	6.3851	0.0000	0.0000

Table B.2: Nodes evaluation in experiment 8 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
71	2.68	5.00	0.00	201.1725	9.5951	0.0000	0.0100
72	2.68	5.00	1.25	256.4445	9.9646	0.0000	0.0900
73	2.68	5.00	2.50	209.2130	7.7253	0.0700	0.0100
74	2.68	5.00	3.75	162.0000	6.9084	0.0000	0.0000
75	2.68	5.00	5.00	228.3345	5.5905	0.0200	0.0000
76	3.84	0.00	0.00	329.5505	12.7095	0.1100	0.1800
77	3.84	0.00	1.25	323.8325	11.6773	0.5000	0.1300
78	3.84	0.00	2.50	350.0660	11.3145	0.5100	0.1600
79	3.84	0.00	3.75	397.1165	11.5885	0.4000	0.3200
80	3.84	0.00	5.00	326.3665	10.2932	0.2900	0.2200
81	3.84	1.25	0.00	236.7190	10.9095	0.0400	0.1000
82	3.84	1.25	1.25	319.4860	11.0613	0.5000	0.0700
83	3.84	1.25	2.50	373.8560	10.5196	0.3100	0.1900
84	3.84	1.25	3.75	388.1200	10.3606	0.2000	0.1500
85	3.84	1.25	5.00	388.2970	8.6120	1.0800	0.2500
86	3.84	2.50	0.00	304.5900	11.6285	0.2700	0.1600
87	3.84	2.50	1.25	332.6860	11.3211	0.0000	0.1300
88	3.84	2.50	2.50	300.3250	9.2026	1.3500	0.0200
89	3.84	2.50	3.75	395.8605	8.6819	0.9400	0.2300
90	3.84	2.50	5.00	307.8020	8.3046	0.5100	0.0500
91	3.84	3.75	0.00	225.6935	10.1077	0.1300	0.0500
92	3.84	3.75	1.25	287.4290	10.6159	0.2700	0.0300
93	3.84	3.75	2.50	381.6755	9.8166	0.7200	0.2100
94	3.84	3.75	3.75	272.3345	7.6991	0.5600	0.0900
95	3.84	3.75	5.00	266.2560	6.8857	1.1800	0.0300
96	3.84	5.00	0.00	238.6480	10.0253	0.0000	0.0500
97	3.84	5.00	1.25	277.3555	8.9212	0.4200	0.0500
98	3.84	5.00	2.50	229.5970	8.5263	0.0500	0.0000
99	3.84	5.00	3.75	233.2815	7.3559	0.4500	0.0000
100	3.84	5.00	5.00	230.0000	7.2600	0.0600	0.0200
101	5.00	0.00	0.00	327.3410	12.3089	0.8000	0.1800
102	5.00	0.00	1.25	389.2670	12.1888	0.6700	0.2200
103	5.00	0.00	2.50	368.3235	12.1825	1.1700	0.2100
104	5.00	0.00	3.75	416.8900	10.9381	0.7000	0.2900
105	5.00	0.00	5.00	422.8370	11.6036	1.8200	0.2700
106	5.00	1.25	0.00	249.6875	10.7982	0.5700	0.0500
107	5.00	1.25	1.25	314.1615	10.7052	0.4400	0.0900
108	5.00	1.25	2.50	317.1330	9.9559	2.2300	0.1100
109	5.00	1.25	3.75	336.7115	10.4970	0.6400	0.2000
110	5.00	1.25	5.00	436.5975	9.6576	0.8000	0.2600

Table B.2: Nodes evaluation in experiment 8 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
111	5.00	2.50	0.00	249.9215	10.6030	0.4100	0.0300
112	5.00	2.50	1.25	297.4430	10.4471	0.0000	0.0600
113	5.00	2.50	2.50	358.2790	10.2209	0.8200	0.1100
114	5.00	2.50	3.75	380.5115	9.6251	3.3300	0.1700
115	5.00	2.50	5.00	358.3100	9.0831	0.7300	0.1200
116	5.00	3.75	0.00	279.3745	11.7875	1.3900	0.0800
117	5.00	3.75	1.25	261.6815	9.8773	0.5000	0.0700
118	5.00	3.75	2.50	336.8765	9.9028	0.4700	0.0600
119	5.00	3.75	3.75	342.5875	8.1286	5.0200	0.1300
120	5.00	3.75	5.00	397.9225	8.7945	2.4500	0.1000
121	5.00	5.00	0.00	268.5595	11.5279	0.7000	0.1000
122	5.00	5.00	1.25	327.0325	10.3172	1.2600	0.1600
123	5.00	5.00	2.50	328.6910	8.9881	1.1600	0.1000
124	5.00	5.00	3.75	306.9280	7.7359	0.8100	0.0700
125	5.00	5.00	5.00	298.7190	7.7199	0.9100	0.0600

B.3 Nodes for experiment 9

In the Table B.3 the evaluation obtained by each generated node after 10 repetitions in the experiment 9 is showed. The evaluation of all the nodes takes 5.8717 days of computational time.

Table B.3: Nodes evaluation in experiment 9

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
1	0.36	0.00	0.00	267.1213	10.8298	75.0450	0.0900
2	0.36	0.00	1.25	288.2560	10.9226	76.5500	0.0950
3	0.36	0.00	2.50	319.7035	10.9643	78.0900	0.1450
4	0.36	0.00	3.75	327.3665	11.1712	72.5200	0.1900
5	0.36	0.00	5.00	374.0782	11.7054	69.1100	0.2750
6	0.36	1.25	0.00	260.1868	11.0402	58.7900	0.1050
7	0.36	1.25	1.25	248.1368	9.4942	72.7150	0.0550
8	0.36	1.25	2.50	311.3928	9.6237	68.4400	0.1100
9	0.36	1.25	3.75	244.4290	8.4278	49.3700	0.0400
10	0.36	1.25	5.00	305.1750	10.4154	50.6600	0.1750
11	0.36	2.50	0.00	235.9830	9.9528	53.9350	0.0600
12	0.36	2.50	1.25	192.3935	7.2186	45.5500	0.0250
13	0.36	2.50	2.50	217.6952	8.0073	20.7950	0.0100
14	0.36	2.50	3.75	207.5055	7.0780	17.6900	0.0050
15	0.36	2.50	5.00	254.6560	9.9986	22.4000	0.0700
16	0.36	3.75	0.00	232.9853	9.5285	58.6550	0.0900
17	0.36	3.75	1.25	217.6455	7.1751	27.0900	0.0400
18	0.36	3.75	2.50	194.8213	6.4885	16.3800	0.0500
19	0.36	3.75	3.75	185.7485	6.3174	9.8550	0.0650
20	0.36	3.75	5.00	191.4015	5.6944	8.8400	0.0000
21	0.36	5.00	0.00	224.7188	9.4535	54.1350	0.0600
22	0.36	5.00	1.25	258.7430	7.2655	22.7850	0.1100
23	0.36	5.00	2.50	184.0627	5.0301	11.7500	0.0200
24	0.36	5.00	3.75	173.1733	3.2716	3.2400	0.0000
25	0.36	5.00	5.00	171.9335	5.1518	5.5200	0.0000
26	1.52	0.00	0.00	270.5398	10.6996	0.0050	0.1000
27	1.52	0.00	1.25	304.1470	11.3981	0.0000	0.1350
28	1.52	0.00	2.50	366.3935	12.3335	0.0000	0.2450
29	1.52	0.00	3.75	436.6927	12.7502	0.0000	0.4250
30	1.52	0.00	5.00	386.6597	12.9275	0.0050	0.3400

Table B.3: Nodes evaluation in experiment 9 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
31	1.52	1.25	0.00	277.5043	10.9315	0.0200	0.1450
32	1.52	1.25	1.25	369.6015	11.4129	0.0000	0.1400
33	1.52	1.25	2.50	385.5740	11.4293	0.0100	0.2450
34	1.52	1.25	3.75	370.5162	10.6932	0.0000	0.2000
35	1.52	1.25	5.00	325.1513	11.7649	0.0000	0.1850
36	1.52	2.50	0.00	244.1328	9.8106	0.0100	0.0750
37	1.52	2.50	1.25	296.5862	9.7082	0.0000	0.0850
38	1.52	2.50	2.50	342.1963	10.1169	0.0050	0.1600
39	1.52	2.50	3.75	359.3720	10.5114	0.0000	0.0800
40	1.52	2.50	5.00	323.4443	10.9041	0.0000	0.1350
41	1.52	3.75	0.00	253.2340	10.4761	0.0150	0.1050
42	1.52	3.75	1.25	313.1698	10.4866	0.0000	0.1100
43	1.52	3.75	2.50	290.0085	9.6015	0.0050	0.0600
44	1.52	3.75	3.75	244.6605	8.7842	0.0000	0.0550
45	1.52	3.75	5.00	222.6203	8.2565	0.0000	0.0100
46	1.52	5.00	0.00	229.0838	9.5644	0.0000	0.0550
47	1.52	5.00	1.25	329.1698	10.0533	0.0050	0.1950
48	1.52	5.00	2.50	244.4783	9.1970	0.0100	0.0050
49	1.52	5.00	3.75	217.8648	7.4773	0.0000	0.0300
50	1.52	5.00	5.00	197.8100	6.9200	0.0000	0.0000
51	2.68	0.00	0.00	290.5910	11.4071	0.0700	0.1300
52	2.68	0.00	1.25	385.9222	13.0169	0.1850	0.1950
53	2.68	0.00	2.50	409.9733	12.7442	0.0850	0.3050
54	2.68	0.00	3.75	395.6855	12.6884	0.0800	0.2850
55	2.68	0.00	5.00	422.5172	13.0651	0.0500	0.2700
56	2.68	1.25	0.00	267.4078	10.6655	0.3100	0.0900
57	2.68	1.25	1.25	353.1232	11.8428	0.2700	0.1200
58	2.68	1.25	2.50	446.5248	11.5437	0.6350	0.3550
59	2.68	1.25	3.75	455.1348	11.4520	0.4350	0.3550
60	2.68	1.25	5.00	436.5780	11.4724	0.4150	0.3050
61	2.68	2.50	0.00	248.0338	10.1313	0.1250	0.0850
62	2.68	2.50	1.25	400.3875	11.3718	0.3300	0.2050
63	2.68	2.50	2.50	483.1938	11.2982	0.1250	0.4300
64	2.68	2.50	3.75	412.3070	10.8465	0.4600	0.1400
65	2.68	2.50	5.00	389.2873	10.1577	0.0800	0.1650
66	2.68	3.75	0.00	257.7407	10.4708	0.0450	0.0750
67	2.68	3.75	1.25	387.9833	11.3196	0.4000	0.2400
68	2.68	3.75	2.50	364.2395	10.3402	0.0850	0.1400
69	2.68	3.75	3.75	307.8743	8.1630	0.2500	0.0000
70	2.68	3.75	5.00	324.3570	9.1340	0.1600	0.0900

Table B.3: Nodes evaluation in experiment 9 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
71	2.68	5.00	0.00	252.8983	10.4694	0.0050	0.0450
72	2.68	5.00	1.25	303.0333	9.3226	0.5950	0.1150
73	2.68	5.00	2.50	319.7208	9.2380	0.1350	0.0100
74	2.68	5.00	3.75	315.1485	9.5298	0.3100	0.0200
75	2.68	5.00	5.00	269.1418	8.2089	0.1350	0.0800
76	3.84	0.00	0.00	349.7350	13.0714	0.1350	0.2200
77	3.84	0.00	1.25	424.6307	13.9101	1.8350	0.3500
78	3.84	0.00	2.50	449.4638	13.5422	2.1650	0.4200
79	3.84	0.00	3.75	464.0005	13.3567	1.7950	0.4450
80	3.84	0.00	5.00	464.0712	13.0909	2.1150	0.4250
81	3.84	1.25	0.00	267.7947	11.3634	0.7900	0.1000
82	3.84	1.25	1.25	397.0715	14.0715	4.3400	0.3150
83	3.84	1.25	2.50	464.1223	11.9242	3.3650	0.3400
84	3.84	1.25	3.75	504.2672	10.7878	4.4200	0.5650
85	3.84	1.25	5.00	535.7850	10.1182	3.6650	0.5950
86	3.84	2.50	0.00	335.9175	11.9335	3.0450	0.1350
87	3.84	2.50	1.25	445.1413	11.5270	3.5050	0.3600
88	3.84	2.50	2.50	489.5065	11.1255	3.5350	0.4200
89	3.84	2.50	3.75	508.4585	9.9286	5.5900	0.5450
90	3.84	2.50	5.00	531.2912	9.7890	3.2050	0.4850
91	3.84	3.75	0.00	258.1225	11.1512	1.3250	0.0600
92	3.84	3.75	1.25	395.4092	11.0616	0.9250	0.1900
93	3.84	3.75	2.50	448.3570	10.9359	1.7750	0.3300
94	3.84	3.75	3.75	482.4632	10.0418	2.2950	0.3550
95	3.84	3.75	5.00	419.2392	8.9253	1.8750	0.1550
96	3.84	5.00	0.00	259.5948	10.7211	0.8450	0.0700
97	3.84	5.00	1.25	325.0670	10.6639	1.1850	0.0900
98	3.84	5.00	2.50	421.6543	9.9504	2.9150	0.2800
99	3.84	5.00	3.75	363.4115	9.6652	1.3850	0.1900
100	3.84	5.00	5.00	416.5635	8.0389	3.0950	0.3450
101	5.00	0.00	0.00	392.1488	13.9356	3.8700	0.3600
102	5.00	0.00	1.25	466.8198	14.7760	5.4900	0.4750
103	5.00	0.00	2.50	469.6328	13.4620	6.9150	0.4900
104	5.00	0.00	3.75	458.5892	13.9523	7.1650	0.4700
105	5.00	0.00	5.00	474.6888	13.0210	6.4550	0.4850
106	5.00	1.25	0.00	361.8567	13.3204	4.7900	0.2850
107	5.00	1.25	1.25	459.6205	12.5678	7.6200	0.4300
108	5.00	1.25	2.50	413.4757	12.3442	4.3000	0.3400
109	5.00	1.25	3.75	506.4275	12.3298	9.1150	0.5600
110	5.00	1.25	5.00	532.4378	10.3079	4.8200	0.6200

Table B.3: Nodes evaluation in experiment 9 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
111	5.00	2.50	0.00	404.7700	12.9592	5.5550	0.3550
112	5.00	2.50	1.25	391.2782	12.1057	4.1000	0.3700
113	5.00	2.50	2.50	483.1715	11.7873	5.1750	0.5050
114	5.00	2.50	3.75	547.8858	9.5752	8.0950	0.7250
115	5.00	2.50	5.00	545.6515	9.8192	6.8500	0.6550
116	5.00	3.75	0.00	360.5038	12.6101	4.2700	0.2850
117	5.00	3.75	1.25	477.4642	12.2858	7.9200	0.5450
118	5.00	3.75	2.50	543.1425	10.4887	9.7100	0.7150
119	5.00	3.75	3.75	570.2612	10.1163	11.9350	0.7300
120	5.00	3.75	5.00	528.7300	9.4765	6.8300	0.5750
121	5.00	5.00	0.00	303.4710	12.2346	1.5950	0.2050
122	5.00	5.00	1.25	448.8592	11.0255	3.8550	0.4050
123	5.00	5.00	2.50	460.4800	10.1418	7.2300	0.4950
124	5.00	5.00	3.75	446.0468	11.3208	3.7800	0.4500
125	5.00	5.00	5.00	472.9503	10.0596	8.0650	0.5500

B.4 Nodes for experiment 10

In the Table B.4 the evaluation obtained by each generated node after 10 repetitions in the experiment 10 is showed. The evaluation of all the nodes takes 1.8453 days of computational time.

Table B.4: Nodes evaluation in experiment 10

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
1	0.36	0.00	0.00	299.9400	12.5102	15.9400	0.1400
2	0.36	0.00	1.25	316.8290	12.4435	19.9400	0.1800
3	0.36	0.00	2.50	318.5540	11.6492	23.1600	0.2000
4	0.36	0.00	3.75	411.9840	9.9089	29.2200	0.3400
5	0.36	0.00	5.00	377.1980	8.4432	28.0800	0.2600
6	0.36	1.25	0.00	285.6960	11.3746	11.4600	0.1600
7	0.36	1.25	1.25	286.6550	10.4908	8.6800	0.1800
8	0.36	1.25	2.50	279.7680	8.6380	10.3200	0.2000
9	0.36	1.25	3.75	345.7880	3.4359	7.1600	0.2000
10	0.36	1.25	5.00	395.0110	3.4977	5.7800	0.2600
11	0.36	2.50	0.00	277.0810	11.3557	8.9600	0.1400
12	0.36	2.50	1.25	246.4160	6.8345	7.7200	0.0600
13	0.36	2.50	2.50	254.7910	4.5026	3.2400	0.1200
14	0.36	2.50	3.75	309.7360	1.7333	2.3800	0.1000
15	0.36	2.50	5.00	184.4030	2.1979	1.2800	0.0000
16	0.36	3.75	0.00	283.4370	11.6117	10.6400	0.1400
17	0.36	3.75	1.25	344.1410	10.7778	6.9600	0.1600
18	0.36	3.75	2.50	278.1130	5.1093	4.1800	0.1200
19	0.36	3.75	3.75	228.7680	2.2270	1.3400	0.1000
20	0.36	3.75	5.00	242.9650	2.8731	1.0200	0.1000
21	0.36	5.00	0.00	321.7070	12.4512	9.3600	0.2200
22	0.36	5.00	1.25	298.4080	8.9259	6.9400	0.2000
23	0.36	5.00	2.50	253.3780	4.2059	3.3600	0.1400
24	0.36	5.00	3.75	310.4880	4.0888	1.6800	0.1200
25	0.36	5.00	5.00	284.5440	2.7889	1.0200	0.1000
26	1.52	0.00	0.00	293.2810	12.0483	0.0000	0.1600
27	1.52	0.00	1.25	384.7470	13.5040	0.0000	0.3200
28	1.52	0.00	2.50	377.3260	11.9575	0.0000	0.2600
29	1.52	0.00	3.75	381.2400	11.1821	0.0000	0.3600
30	1.52	0.00	5.00	374.3390	9.0160	0.0000	0.2600

Table B.4: Nodes evaluation in experiment 10 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
31	1.52	1.25	0.00	288.1900	12.1456	0.0000	0.1000
32	1.52	1.25	1.25	309.1290	10.8358	0.0000	0.1400
33	1.52	1.25	2.50	344.0650	7.3302	0.0000	0.0200
34	1.52	1.25	3.75	302.3350	7.4041	0.0000	0.1200
35	1.52	1.25	5.00	284.3820	4.4003	0.0000	0.1800
36	1.52	2.50	0.00	331.3360	12.5451	0.0000	0.1800
37	1.52	2.50	1.25	394.5020	11.1746	0.0000	0.2600
38	1.52	2.50	2.50	330.2020	6.4296	0.0000	0.1400
39	1.52	2.50	3.75	229.3820	3.2359	0.0000	0.0200
40	1.52	2.50	5.00	304.4720	3.4278	0.0000	0.1000
41	1.52	3.75	0.00	309.6710	11.6464	0.0000	0.1200
42	1.52	3.75	1.25	331.8070	10.0223	0.0000	0.1400
43	1.52	3.75	2.50	282.4240	6.3699	0.0000	0.0600
44	1.52	3.75	3.75	276.2250	3.8590	0.0000	0.2200
45	1.52	3.75	5.00	278.8050	5.6153	0.0000	0.1000
46	1.52	5.00	0.00	261.8990	10.9411	0.0000	0.1000
47	1.52	5.00	1.25	264.9750	9.6526	0.0000	0.1000
48	1.52	5.00	2.50	313.5710	8.3697	0.0000	0.1600
49	1.52	5.00	3.75	251.2500	4.2758	0.0000	0.0000
50	1.52	5.00	5.00	260.5210	3.8144	0.3600	0.0000
51	2.68	0.00	0.00	293.0600	11.7489	0.0000	0.1400
52	2.68	0.00	1.25	373.6820	12.4944	0.0200	0.2400
53	2.68	0.00	2.50	374.1660	10.6716	0.0600	0.2600
54	2.68	0.00	3.75	362.6450	10.4439	0.0000	0.2200
55	2.68	0.00	5.00	421.5870	10.9305	0.0000	0.4000
56	2.68	1.25	0.00	291.9920	12.4512	0.0000	0.1400
57	2.68	1.25	1.25	291.1650	12.2431	0.0000	0.0400
58	2.68	1.25	2.50	272.7310	10.1253	0.0000	0.1200
59	2.68	1.25	3.75	258.6170	6.2267	0.0000	0.0600
60	2.68	1.25	5.00	255.2920	5.1846	0.0000	0.0600
61	2.68	2.50	0.00	315.1920	12.3013	0.0000	0.1600
62	2.68	2.50	1.25	256.5970	9.1344	0.0000	0.0600
63	2.68	2.50	2.50	347.4250	7.5259	0.0000	0.2200
64	2.68	2.50	3.75	211.6220	5.5757	0.0000	0.0000
65	2.68	2.50	5.00	195.8300	3.4990	0.0000	0.0000
66	2.68	3.75	0.00	265.7800	11.2395	0.0000	0.1200
67	2.68	3.75	1.25	293.4050	9.5180	0.0000	0.0600
68	2.68	3.75	2.50	274.4380	6.7262	0.0000	0.0600
69	2.68	3.75	3.75	195.0920	4.7740	0.0000	0.0000
70	2.68	3.75	5.00	283.3650	4.7855	0.0000	0.2000

Table B.4: Nodes evaluation in experiment 10 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
71	2.68	5.00	0.00	307.6550	12.0780	0.0000	0.1600
72	2.68	5.00	1.25	307.9840	9.9940	0.0000	0.1000
73	2.68	5.00	2.50	325.0240	7.7681	0.0000	0.0800
74	2.68	5.00	3.75	252.5790	4.6011	0.0000	0.0000
75	2.68	5.00	5.00	325.0020	4.3283	0.0000	0.1600
76	3.84	0.00	0.00	370.0350	12.5465	0.0000	0.1200
77	3.84	0.00	1.25	349.9750	12.9603	0.0000	0.2200
78	3.84	0.00	2.50	378.1680	10.2515	0.0000	0.2200
79	3.84	0.00	3.75	471.5240	11.9554	0.0000	0.4800
80	3.84	0.00	5.00	400.2780	8.5786	0.0000	0.2400
81	3.84	1.25	0.00	242.3610	10.7808	0.0000	0.0400
82	3.84	1.25	1.25	337.6310	11.6872	0.0000	0.2000
83	3.84	1.25	2.50	320.7640	9.3862	0.0000	0.1200
84	3.84	1.25	3.75	325.8960	8.7456	0.0000	0.0600
85	3.84	1.25	5.00	427.4340	5.8895	1.2800	0.3200
86	3.84	2.50	0.00	314.3480	11.6590	0.0000	0.2200
87	3.84	2.50	1.25	312.6230	10.9539	0.0000	0.1600
88	3.84	2.50	2.50	365.0530	9.7166	1.2800	0.2200
89	3.84	2.50	3.75	265.1830	6.3521	0.4600	0.0200
90	3.84	2.50	5.00	302.8700	4.7323	1.0200	0.1600
91	3.84	3.75	0.00	294.0630	11.6735	0.0000	0.1600
92	3.84	3.75	1.25	289.4170	10.2235	0.0000	0.1000
93	3.84	3.75	2.50	306.4530	7.6083	0.1600	0.0400
94	3.84	3.75	3.75	386.9910	5.5574	1.1000	0.2000
95	3.84	3.75	5.00	249.9690	5.4047	0.1400	0.0000
96	3.84	5.00	0.00	268.8950	10.4629	0.0000	0.0400
97	3.84	5.00	1.25	302.7840	9.0182	0.0000	0.1600
98	3.84	5.00	2.50	253.7520	7.0398	0.0000	0.0800
99	3.84	5.00	3.75	208.6010	6.7496	0.0000	0.0400
100	3.84	5.00	5.00	362.4210	5.0618	1.8800	0.2000
101	5.00	0.00	0.00	317.8690	12.8542	0.0000	0.1000
102	5.00	0.00	1.25	376.2870	12.0587	0.1000	0.2200
103	5.00	0.00	2.50	449.4350	10.8182	0.0000	0.2800
104	5.00	0.00	3.75	402.0830	11.2195	0.0000	0.2800
105	5.00	0.00	5.00	408.7090	8.7876	0.7800	0.2000
106	5.00	1.25	0.00	274.3440	11.6440	0.0000	0.1600
107	5.00	1.25	1.25	359.5160	10.9200	0.0000	0.1800
108	5.00	1.25	2.50	432.4940	9.3142	1.1800	0.4600
109	5.00	1.25	3.75	393.1270	8.9580	0.1600	0.3600
110	5.00	1.25	5.00	380.5630	8.1914	0.2400	0.2200

Table B.4: Nodes evaluation in experiment 10 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
111	5.00	2.50	0.00	277.0300	11.4256	0.0000	0.1000
112	5.00	2.50	1.25	370.0020	11.1784	0.0600	0.1800
113	5.00	2.50	2.50	376.7760	10.3550	0.0000	0.2000
114	5.00	2.50	3.75	435.3420	8.4369	0.1600	0.3800
115	5.00	2.50	5.00	448.4700	6.2972	1.5600	0.4000
116	5.00	3.75	0.00	314.3620	12.1880	0.2000	0.1600
117	5.00	3.75	1.25	280.0520	10.0053	0.0000	0.0400
118	5.00	3.75	2.50	341.8610	8.5659	0.5400	0.1600
119	5.00	3.75	3.75	370.9510	6.9029	0.0000	0.2400
120	5.00	3.75	5.00	423.5240	6.1466	0.0000	0.3000
121	5.00	5.00	0.00	257.5040	11.9783	0.0000	0.0800
122	5.00	5.00	1.25	316.8250	9.8933	0.0000	0.1600
123	5.00	5.00	2.50	363.2690	9.2284	0.2400	0.2200
124	5.00	5.00	3.75	372.5160	6.6977	0.7000	0.1200
125	5.00	5.00	5.00	323.0250	7.1166	0.0000	0.1800

B.5 Nodes for experiment 11

In the Table B.5 the evaluation obtained by each generated node after 10 repetitions in the experiment 11 is showed. The evaluation of all the nodes takes 4.5991 days of computational time.

Table B.5: Nodes evaluation in experiment 11

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
1	0.36	0.00	0.00	282.2120	11.1506	42.1100	0.1300
2	0.36	0.00	1.25	349.8885	12.8718	32.3900	0.1600
3	0.36	0.00	2.50	377.1570	11.9641	41.5900	0.2500
4	0.36	0.00	3.75	400.0120	11.5403	44.2000	0.3100
5	0.36	0.00	5.00	414.7865	11.6664	48.4800	0.3200
6	0.36	1.25	0.00	331.1600	12.4970	24.0900	0.2100
7	0.36	1.25	1.25	330.2880	10.8336	31.5000	0.1500
8	0.36	1.25	2.50	282.8235	8.7206	33.0700	0.0900
9	0.36	1.25	3.75	289.2030	7.2792	20.3800	0.0600
10	0.36	1.25	5.00	269.2955	8.5845	17.8300	0.0100
11	0.36	2.50	0.00	329.4415	13.2156	18.0900	0.1400
12	0.36	2.50	1.25	290.9365	9.7671	20.1300	0.1200
13	0.36	2.50	2.50	221.9730	4.9682	10.6300	0.0400
14	0.36	2.50	3.75	181.0275	2.1076	2.6900	0.0000
15	0.36	2.50	5.00	293.5845	5.4534	9.5700	0.1400
16	0.36	3.75	0.00	290.0270	10.9331	25.3200	0.1100
17	0.36	3.75	1.25	255.6295	8.2801	18.0400	0.0700
18	0.36	3.75	2.50	221.9690	2.5562	2.6700	0.0000
19	0.36	3.75	3.75	276.1370	4.0590	4.9200	0.0500
20	0.36	3.75	5.00	326.0000	2.8851	3.0400	0.1300
21	0.36	5.00	0.00	262.3215	11.4009	22.4500	0.1000
22	0.36	5.00	1.25	260.6710	7.7186	13.5500	0.0900
23	0.36	5.00	2.50	243.5000	4.0857	5.5200	0.0100
24	0.36	5.00	3.75	230.6780	3.0537	2.0700	0.1000
25	0.36	5.00	5.00	252.6780	3.0003	2.1100	0.1000
26	1.52	0.00	0.00	314.4540	12.4839	0.0000	0.2000
27	1.52	0.00	1.25	335.4425	12.4813	0.0000	0.2200
28	1.52	0.00	2.50	394.6030	12.8622	0.0000	0.3000
29	1.52	0.00	3.75	419.6180	12.5951	0.0000	0.3500
30	1.52	0.00	5.00	458.3465	13.0137	0.0000	0.4000

Table B.5: Nodes evaluation in experiment 11 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
31	1.52	1.25	0.00	315.0970	12.3257	0.0000	0.1400
32	1.52	1.25	1.25	343.1160	11.5626	0.0000	0.1500
33	1.52	1.25	2.50	424.1555	13.1043	0.0000	0.3700
34	1.52	1.25	3.75	416.6280	10.1635	0.0000	0.2900
35	1.52	1.25	5.00	372.8385	10.3787	0.0000	0.1400
36	1.52	2.50	0.00	301.4560	11.3148	0.0000	0.1600
37	1.52	2.50	1.25	344.0950	10.9643	0.0100	0.1800
38	1.52	2.50	2.50	269.1450	9.9487	0.0000	0.0500
39	1.52	2.50	3.75	308.2765	7.4838	0.0000	0.1000
40	1.52	2.50	5.00	332.4470	9.5512	0.0000	0.2400
41	1.52	3.75	0.00	268.3975	10.2705	0.0000	0.1200
42	1.52	3.75	1.25	266.0800	9.6632	0.0000	0.1000
43	1.52	3.75	2.50	338.0925	8.8854	0.0000	0.1300
44	1.52	3.75	3.75	204.8160	3.7293	0.0000	0.0000
45	1.52	3.75	5.00	212.0465	4.7682	0.0000	0.0000
46	1.52	5.00	0.00	298.1535	11.8254	0.0100	0.1500
47	1.52	5.00	1.25	343.8570	10.0124	0.0000	0.1700
48	1.52	5.00	2.50	289.9135	5.9992	0.0000	0.1400
49	1.52	5.00	3.75	249.2770	4.0682	0.0000	0.0000
50	1.52	5.00	5.00	290.4910	4.0618	0.0000	0.0000
51	2.68	0.00	0.00	345.3845	12.6529	0.0000	0.2000
52	2.68	0.00	1.25	369.0715	12.9426	0.1000	0.2400
53	2.68	0.00	2.50	428.2065	13.1370	0.0000	0.3400
54	2.68	0.00	3.75	477.5120	11.4781	0.1400	0.4000
55	2.68	0.00	5.00	442.3920	11.8098	0.0100	0.3400
56	2.68	1.25	0.00	322.4470	12.6268	0.0600	0.2500
57	2.68	1.25	1.25	334.6280	11.3070	0.0700	0.1300
58	2.68	1.25	2.50	354.3115	10.3718	0.0500	0.1100
59	2.68	1.25	3.75	470.2030	10.9902	0.1500	0.4400
60	2.68	1.25	5.00	438.4370	8.7841	0.2700	0.3200
61	2.68	2.50	0.00	282.2685	10.6866	0.0000	0.1100
62	2.68	2.50	1.25	323.1405	11.0765	0.0000	0.1300
63	2.68	2.50	2.50	399.8345	9.2133	0.3400	0.3000
64	2.68	2.50	3.75	347.7150	8.1424	0.0400	0.1100
65	2.68	2.50	5.00	423.3165	9.2603	0.1200	0.2500
66	2.68	3.75	0.00	298.4045	11.4680	0.0300	0.1800
67	2.68	3.75	1.25	344.3370	10.5002	0.0000	0.2000
68	2.68	3.75	2.50	352.1305	8.7849	0.0000	0.1000
69	2.68	3.75	3.75	284.2900	8.1058	0.0100	0.0000
70	2.68	3.75	5.00	401.1615	6.3537	0.3000	0.3200

Table B.5: Nodes evaluation in experiment 11 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
71	2.68	5.00	0.00	276.9775	10.9749	0.0000	0.1600
72	2.68	5.00	1.25	317.0965	9.1621	0.0300	0.0800
73	2.68	5.00	2.50	208.6360	7.5986	0.0300	0.0000
74	2.68	5.00	3.75	258.7290	6.3791	0.0100	0.1000
75	2.68	5.00	5.00	291.2130	5.4994	0.0200	0.1800
76	3.84	0.00	0.00	304.4950	11.5282	0.0000	0.1000
77	3.84	0.00	1.25	396.0565	13.0750	0.1100	0.2600
78	3.84	0.00	2.50	452.8140	13.2668	0.2600	0.4300
79	3.84	0.00	3.75	454.4915	12.2172	0.5600	0.3700
80	3.84	0.00	5.00	500.6905	12.9126	0.7400	0.5100
81	3.84	1.25	0.00	305.3020	11.5228	0.3100	0.0600
82	3.84	1.25	1.25	340.4465	11.8400	0.6200	0.1500
83	3.84	1.25	2.50	411.6275	11.3272	0.0700	0.2700
84	3.84	1.25	3.75	419.5250	10.8021	0.8800	0.2800
85	3.84	1.25	5.00	406.8945	8.1100	0.3300	0.2200
86	3.84	2.50	0.00	282.3665	11.5876	0.0900	0.1000
87	3.84	2.50	1.25	349.1055	11.5907	0.2200	0.2300
88	3.84	2.50	2.50	387.7325	9.9961	0.0000	0.2500
89	3.84	2.50	3.75	403.4235	11.3642	0.2500	0.1900
90	3.84	2.50	5.00	405.1025	8.1674	1.7700	0.2400
91	3.84	3.75	0.00	279.7265	11.7198	0.0100	0.0900
92	3.84	3.75	1.25	355.4595	10.6346	0.3600	0.0900
93	3.84	3.75	2.50	397.9595	9.8469	0.1800	0.2000
94	3.84	3.75	3.75	444.2385	9.2873	0.3500	0.3100
95	3.84	3.75	5.00	433.3845	7.5452	0.2900	0.2700
96	3.84	5.00	0.00	308.1230	11.6512	0.1200	0.1400
97	3.84	5.00	1.25	352.5790	10.3138	0.1900	0.1300
98	3.84	5.00	2.50	378.4530	9.0926	0.5700	0.2100
99	3.84	5.00	3.75	378.6195	7.5700	2.8300	0.2600
100	3.84	5.00	5.00	377.4215	6.4474	0.9700	0.2800
101	5.00	0.00	0.00	357.4810	12.7231	0.0400	0.1800
102	5.00	0.00	1.25	433.3745	13.0702	0.6900	0.3600
103	5.00	0.00	2.50	468.8730	13.0202	1.0900	0.4500
104	5.00	0.00	3.75	478.8045	13.0135	2.0400	0.4700
105	5.00	0.00	5.00	527.7375	13.5313	2.8500	0.6000
106	5.00	1.25	0.00	347.1125	12.2343	0.9800	0.1500
107	5.00	1.25	1.25	380.5115	11.8111	0.4100	0.2300
108	5.00	1.25	2.50	429.9455	10.5956	1.3800	0.4000
109	5.00	1.25	3.75	473.5530	11.0151	2.8400	0.4000
110	5.00	1.25	5.00	518.1210	8.6467	2.4400	0.5700

Table B.5: Nodes evaluation in experiment 11 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
111	5.00	2.50	0.00	325.1725	12.2598	0.6100	0.1300
112	5.00	2.50	1.25	381.2895	11.6325	2.3600	0.2300
113	5.00	2.50	2.50	475.7100	12.1457	0.9500	0.4500
114	5.00	2.50	3.75	469.5685	9.0061	3.5000	0.4600
115	5.00	2.50	5.00	475.6055	9.7166	3.7100	0.3400
116	5.00	3.75	0.00	279.4620	11.4065	0.0600	0.0800
117	5.00	3.75	1.25	362.2410	10.1234	0.6800	0.1900
118	5.00	3.75	2.50	459.7160	9.7145	2.0400	0.3900
119	5.00	3.75	3.75	442.2425	8.2634	1.8400	0.3400
120	5.00	3.75	5.00	495.3780	8.3016	1.1200	0.5400
121	5.00	5.00	0.00	256.7960	10.7812	0.3300	0.0400
122	5.00	5.00	1.25	387.8190	10.8514	1.1800	0.1700
123	5.00	5.00	2.50	469.3020	8.1639	6.3600	0.3500
124	5.00	5.00	3.75	466.2425	9.2899	2.9700	0.3900
125	5.00	5.00	5.00	355.1740	7.1385	2.9900	0.1800

B.6 Nodes for experiment 12

In the Table B.6 the evaluation obtained by each generated node after 10 repetitions in the experiment 12 is showed. The evaluation of all the nodes takes 9.4936 days of computational time.

Table B.6: Nodes evaluation in experiment 12

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
1	0.36	0.00	0.00	305.3705	11.8529	65.5750	0.1450
2	0.36	0.00	1.25	317.1152	11.8411	70.4650	0.1400
3	0.36	0.00	2.50	430.0398	13.9529	57.9100	0.4100
4	0.36	0.00	3.75	386.7500	12.3238	72.5850	0.2700
5	0.36	0.00	5.00	408.7912	12.4396	61.0350	0.3300
6	0.36	1.25	0.00	282.4935	11.1752	61.1800	0.0900
7	0.36	1.25	1.25	298.5092	10.6169	71.2300	0.1350
8	0.36	1.25	2.50	328.8240	10.2922	55.9450	0.1550
9	0.36	1.25	3.75	295.0122	10.2396	42.4300	0.1400
10	0.36	1.25	5.00	305.2620	9.4978	53.7200	0.2000
11	0.36	2.50	0.00	285.4577	11.1221	47.7250	0.1100
12	0.36	2.50	1.25	353.5460	11.0071	36.7350	0.2200
13	0.36	2.50	2.50	296.2240	5.6017	21.1750	0.1900
14	0.36	2.50	3.75	294.7505	7.7720	19.8200	0.1600
15	0.36	2.50	5.00	270.4620	7.0694	17.5300	0.0950
16	0.36	3.75	0.00	268.7240	10.9298	48.1550	0.1150
17	0.36	3.75	1.25	305.1513	9.6896	47.0200	0.1200
18	0.36	3.75	2.50	284.3415	5.8995	25.4350	0.1500
19	0.36	3.75	3.75	252.0093	4.6997	9.5550	0.0000
20	0.36	3.75	5.00	277.2173	4.9174	8.7450	0.0000
21	0.36	5.00	0.00	234.0215	9.5935	55.1800	0.0800
22	0.36	5.00	1.25	307.1410	7.0502	24.1600	0.1000
23	0.36	5.00	2.50	200.1877	4.2316	8.7500	0.0400
24	0.36	5.00	3.75	147.8287	3.3053	2.7000	0.0000
25	0.36	5.00	5.00	180.0292	3.3017	3.5850	0.0000
26	1.52	0.00	0.00	328.0260	12.4681	0.0000	0.1750
27	1.52	0.00	1.25	382.1208	12.5036	0.0000	0.2000
28	1.52	0.00	2.50	413.8515	13.1335	0.0000	0.2450
29	1.52	0.00	3.75	447.3753	12.5550	0.0000	0.3850
30	1.52	0.00	5.00	472.6420	13.3501	0.0000	0.4950

Table B.6: Nodes evaluation in experiment 12 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
31	1.52	1.25	0.00	295.0740	11.7028	0.0050	0.1350
32	1.52	1.25	1.25	378.4845	11.4512	0.0050	0.1900
33	1.52	1.25	2.50	391.5465	12.3955	0.0000	0.1850
34	1.52	1.25	3.75	455.2953	12.2800	0.0000	0.3100
35	1.52	1.25	5.00	464.0193	11.6168	0.0000	0.2850
36	1.52	2.50	0.00	292.3330	11.4123	0.0050	0.1450
37	1.52	2.50	1.25	425.3998	12.8827	0.0000	0.3150
38	1.52	2.50	2.50	340.7947	10.1739	0.0000	0.1800
39	1.52	2.50	3.75	351.6463	11.6578	0.0000	0.2050
40	1.52	2.50	5.00	377.6383	10.5685	0.0000	0.2000
41	1.52	3.75	0.00	262.5780	10.6779	0.0150	0.0800
42	1.52	3.75	1.25	377.8408	11.1852	0.0050	0.1650
43	1.52	3.75	2.50	387.5415	9.8185	0.0050	0.2450
44	1.52	3.75	3.75	312.5938	9.8204	0.0050	0.1450
45	1.52	3.75	5.00	281.5328	7.5935	0.1450	0.1150
46	1.52	5.00	0.00	288.7950	11.3045	0.0100	0.1200
47	1.52	5.00	1.25	304.6708	9.8628	0.0100	0.1150
48	1.52	5.00	2.50	305.4218	9.4638	0.0150	0.0150
49	1.52	5.00	3.75	322.5203	7.5137	0.2450	0.1200
50	1.52	5.00	5.00	261.2333	7.6386	0.0000	0.0400
51	2.68	0.00	0.00	384.4188	14.2364	0.0100	0.3100
52	2.68	0.00	1.25	444.6908	14.0130	0.1650	0.3800
53	2.68	0.00	2.50	481.2625	13.6988	0.4250	0.5000
54	2.68	0.00	3.75	485.2478	13.4761	0.4450	0.4450
55	2.68	0.00	5.00	503.5205	13.2125	0.2400	0.5250
56	2.68	1.25	0.00	325.9683	11.7924	0.0800	0.1550
57	2.68	1.25	1.25	420.9490	13.0644	0.1850	0.3100
58	2.68	1.25	2.50	510.7998	10.9711	0.6800	0.4800
59	2.68	1.25	3.75	554.6905	11.3322	0.9700	0.6600
60	2.68	1.25	5.00	483.5683	11.7564	0.2350	0.4250
61	2.68	2.50	0.00	309.3388	11.6551	0.1200	0.1400
62	2.68	2.50	1.25	436.5150	11.0631	0.5450	0.2950
63	2.68	2.50	2.50	529.7507	11.2900	1.1450	0.6250
64	2.68	2.50	3.75	519.3098	9.2047	0.8950	0.5600
65	2.68	2.50	5.00	507.5545	9.8856	0.1100	0.3850
66	2.68	3.75	0.00	293.2638	11.1364	0.4000	0.0750
67	2.68	3.75	1.25	364.3215	11.3443	0.2600	0.1450
68	2.68	3.75	2.50	429.0095	10.7540	0.1450	0.2700
69	2.68	3.75	3.75	417.0320	10.5226	0.5650	0.2550
70	2.68	3.75	5.00	451.1075	8.9781	1.0250	0.3350

Table B.6: Nodes evaluation in experiment 12 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
71	2.68	5.00	0.00	275.3220	10.9839	0.1450	0.0600
72	2.68	5.00	1.25	408.4022	10.4592	0.0850	0.2600
73	2.68	5.00	2.50	453.7425	10.0976	0.2750	0.3550
74	2.68	5.00	3.75	420.4510	9.9334	1.8500	0.2750
75	2.68	5.00	5.00	429.5467	9.0538	0.8450	0.2500
76	3.84	0.00	0.00	387.4193	13.5309	1.2100	0.2900
77	3.84	0.00	1.25	493.7718	13.6280	1.4350	0.5300
78	3.84	0.00	2.50	507.5660	12.7432	2.0000	0.5400
79	3.84	0.00	3.75	509.4287	13.1278	1.3750	0.4650
80	3.84	0.00	5.00	493.8152	12.5632	1.4250	0.4550
81	3.84	1.25	0.00	350.3670	12.2948	2.2600	0.1400
82	3.84	1.25	1.25	430.8628	13.0514	3.4250	0.4050
83	3.84	1.25	2.50	505.3293	11.9526	4.0200	0.5850
84	3.84	1.25	3.75	562.6290	10.7801	3.3350	0.7200
85	3.84	1.25	5.00	537.9047	11.0758	2.5300	0.5500
86	3.84	2.50	0.00	355.3400	12.7407	0.8950	0.1950
87	3.84	2.50	1.25	435.6062	11.9307	1.3700	0.2350
88	3.84	2.50	2.50	524.1502	11.4700	2.8850	0.5100
89	3.84	2.50	3.75	574.0870	9.4799	3.6200	0.7900
90	3.84	2.50	5.00	557.2190	8.7721	3.3250	0.7400
91	3.84	3.75	0.00	304.3378	11.4141	1.1150	0.0800
92	3.84	3.75	1.25	499.5885	10.6690	2.4250	0.4750
93	3.84	3.75	2.50	542.5335	9.7995	2.7700	0.7150
94	3.84	3.75	3.75	539.8193	9.4985	5.6650	0.7200
95	3.84	3.75	5.00	541.3010	8.3520	3.3250	0.6750
96	3.84	5.00	0.00	349.6890	12.6306	1.6150	0.1550
97	3.84	5.00	1.25	489.6318	10.7148	1.6500	0.4550
98	3.84	5.00	2.50	510.1505	9.1254	5.1600	0.5950
99	3.84	5.00	3.75	536.9462	9.7904	3.7850	0.6400
100	3.84	5.00	5.00	503.6043	8.5671	5.4600	0.5000
101	5.00	0.00	0.00	443.4103	13.7043	4.4900	0.4600
102	5.00	0.00	1.25	514.3138	13.5308	8.4450	0.6350
103	5.00	0.00	2.50	525.2500	13.2870	8.0350	0.6550
104	5.00	0.00	3.75	546.4540	12.7879	8.2850	0.7650
105	5.00	0.00	5.00	525.1198	13.2873	8.1800	0.6200
106	5.00	1.25	0.00	412.7022	14.2225	5.5650	0.3600
107	5.00	1.25	1.25	485.2123	13.0761	7.1700	0.5550
108	5.00	1.25	2.50	547.1840	12.3831	8.9450	0.7050
109	5.00	1.25	3.75	556.2273	11.2715	9.1800	0.7400
110	5.00	1.25	5.00	570.6868	10.6224	6.5800	0.7850

Table B.6: Nodes evaluation in experiment 12 (continued)

Node	Δr	Δo	Δa	f_1	f_2	f_3	f_4
111	5.00	2.50	0.00	424.4308	14.0671	6.0750	0.3700
112	5.00	2.50	1.25	485.9875	12.9084	8.5400	0.5450
113	5.00	2.50	2.50	543.9778	11.6443	5.6500	0.6850
114	5.00	2.50	3.75	577.4535	10.3853	15.2750	0.8700
115	5.00	2.50	5.00	578.8080	10.5020	9.8900	0.8750
116	5.00	3.75	0.00	381.7152	12.6534	2.2550	0.3000
117	5.00	3.75	1.25	523.5688	11.8490	9.1700	0.6900
118	5.00	3.75	2.50	575.7250	10.0714	15.5250	0.8400
119	5.00	3.75	3.75	572.7835	9.9491	9.2150	0.9050
120	5.00	3.75	5.00	542.0963	9.0343	13.8350	0.7500
121	5.00	5.00	0.00	384.5543	13.5069	4.2000	0.2950
122	5.00	5.00	1.25	513.6340	10.2875	6.5350	0.6300
123	5.00	5.00	2.50	567.9353	9.9905	9.2350	0.8150
124	5.00	5.00	3.75	595.6993	10.1128	8.7050	0.9500
125	5.00	5.00	5.00	575.7138	9.1121	8.3950	0.8850

Bibliography

[1] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, pp. 7–27, Mar 1997.

[2] J. C. Barca and Y. A. Sekercioglu, "Swarm robotics reviewed," *Robotica*, vol. 31, no. 3, p. 345359, 2013.

[3] H. Hamann, *Swarm Robotics: A Formal Approach*. Springer, 1 ed., 01 2018.

[4] B. Zhu, L. Xie, D. Han, X. Meng, and R. Teo, "A survey on recent progress in control of swarm systems," *Science China Information Sciences*, vol. 60, p. 070201, Jun 2017.

[5] S. Fong, S. Deb, and A. Chaudhary, "A review of metaheuristics in robotics," *Comput. Electr. Eng.*, vol. 43, pp. 278–291, apr 2015.

[6] J. Gomes, P. Urbano, and A. L. Christensen, "Evolution of swarm robotics systems with novelty search," *Swarm Intelligence*, vol. 7, pp. 115–144, Sep 2013.

[7] F. Mukhlish, J. Page, and M. Bain, "Evolutionary-learning framework: improving automatic swarm robotics design," *International Journal of Intelligent Unmanned Systems*, vol. 6, no. 4, pp. 197–215, 2018.

[8] M. Dadgar, S. Jafari, and A. Hamzeh, "A pso-based multi-robot cooperation method for target searching in unknown environments," *Neurocomputing*, vol. 177, pp. 62–74, 2016.

[9] Y. Yan, R. Zhang, J. Wang, and J. Li, "Modified pso algorithms with request and reset for leak source localization using multiple robots," *Neurocomputing*, vol. 292, pp. 82–90, 2018.

[10] F. R. Inácio, D. G. Macharet, and L. Chaimowicz, "Pso-based strategy for the segregation of heterogeneous robotic swarms," *Journal of Computational Science*, vol. 31, pp. 86–94, 2019.

[11] R. Fujisawa, S. Dobata, K. Sugawara, and F. Matsuno, "Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance," *Swarm Intelligence*, vol. 8, pp. 227–246, Sep 2014.

[12] P. Suárez, A. Iglesias, and A. Gálvez, "Make robots be bats: specializing robotic swarms to the bat algorithm," *Swarm and Evolutionary Computation*, vol. 44, pp. 113–129, 2019.

[13] Y. Katada, A. Nishiguchi, K. Moriwaki, and R. Watakabe, "Swarm robotic network using lévy flight in target detection problem," *Artificial Life and Robotics*, vol. 21, pp. 295–301, Sep 2016.

[14] L. S. Junior and N. Nedjah, "Wave algorithm applied to collective navigation of robotic swarms," *Applied Soft Computing*, vol. 57, pp. 698–707, 2017.

[15] E. Castello, T. Yamamoto, Y. Nakamura, and H. Ishiguro, "Foraging optimization in swarm robotic systems based on an adaptive response threshold model," *Advanced Robotics*, vol. 28, no. 20, pp. 1343–1356, 2014.

[16] E. Castello, T. Yamamoto, F. D. Libera, W. Liu, A. F. T. Winfield, Y. Nakamura, and H. Ishiguro, "Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach," *Swarm Intelligence*, vol. 10, pp. 1–31, Mar 2016.

[17] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. Ferreira, "Benchmark of swarm robotics distributed techniques in a search task," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 200–213, 2014.

[18] D. Wang, H. Wang, and L. Liu, "Unknown environment exploration of multi-robot system with the fordpso," *Swarm and Evolutionary Computation*, vol. 26, pp. 157–174, 2016.

[19] G. Beni, "From swarm intelligence to swarm robotics," in *Swarm Robotics* (E. Şahin and W. M. Spears, eds.), (Berlin, Heidelberg), pp. 1–9, Springer Berlin Heidelberg, 2005.

[20] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics* (E. Şahin and W. M. Spears, eds.), (Berlin, Heidelberg), pp. 10–20, Springer Berlin Heidelberg, 2005.

[21] S. Camazine, N. R. Franks, J. Sneyd, E. Bonabeau, J.-L. Deneubourg, and G. Theraula, *Self-Organization in Biological Systems*. Princeton, NJ, USA: Princeton University Press, 2001.

[22] V. V. Isaeva, "Self-organization in biological systems," *Biology Bulletin*, vol. 39, pp. 110–118, Apr 2012.

[23] S. Garnier, J. Gautrais, and G. Theraulaz, "The biological principles of swarm intelligence," *Swarm Intelligence*, vol. 1, pp. 3–31, Jun 2007.

[24] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, pp. 1–41, Mar 2013.

[25] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[26] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, pp. 401–420, March 2006.

[27] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, pp. 25–34, aug 1987.

[28] E. Masehian and M. Royan, "Characteristics of and approaches to flocking in swarm robotics," in *Recent Tendency in Aerospace, Robotics, Manufacturing Systems, Energy and Mechanical Engineering*, vol. 841 of *Applied Mechanics and Materials*, pp. 240–249, Trans Tech Publications Ltd, 7 2016.

[29] A. E. "Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, pp. 97–120, Dec 2008.

[30] A. Kumar, S. Sharma, R. Tiwari, and S. Majumdar, "Area exploration by flocking of multi robot," *Procedia Engineering*, vol. 41, pp. 377–382, 2012. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

[31] S. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, pp. 837–855, Aug 2018.

[32] S. Bouabdallah, *Design and control of quadrotors with application to autonomous flying*. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2007.

[33] T. Stirling, J. Roberts, J. Zufferey, and D. Floreano, "Indoor navigation with a swarm of flying robots," in *2012 IEEE International Conference on Robotics and Automation*, pp. 4641–4647, May 2012.

[34] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, pp. 287–300, Nov 2013.

[35] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization," *Autonomous Robots*, vol. 41, pp. 919–944, Apr 2017.

[36] G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szörényi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial

robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3866–3873, Sep. 2014.

[37] A. L. Alfeo, M. G. C. A. Cimino, N. De Francesco, A. Lazzeri, M. Lega, and G. Vaglini, "Swarm coordination of mini-uavs for target search using imperfect sensors," *Intelligent Decision Technologies*, vol. 12, pp. 149–162, Mar 2018.

[38] P. Garcia-Aunon and A. B. Cruz, "Control optimization of an aerial robotic swarm in a search task and its adaptation to different scenarios," *Journal of Computational Science*, vol. 29, pp. 107–118, 2018.

[39] P. Garcia-Aunon, J. J. Roldn, and A. Barrientos, "Monitoring traffic in future cities with aerial swarms: Developing and optimizing a behavior-based surveillance algorithm," *Cognitive Systems Research*, vol. 54, pp. 273–286, 2019.

[40] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[41] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, 2018.

[42] A. L. Alfeo, M. G. Cimino, and G. Vaglini, "Enhancing biologically inspired swarm behavior: Metaheuristics to foster the optimization of uavs coordination in target search," *Computers & Operations Research*, vol. 110, pp. 34–47, 2019.

[43] V. Trianni and M. López-Ibáñez, "Advantages of task-specific multi-objective optimisation in evolutionary robotics," *PLOS ONE*, vol. 10, pp. 1–27, 08 2015.

[44] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, pp. 61–70, 06 2011.

[45] J. A. Duro, R. C. Purshouse, S. Salomon, D. C. Oara, V. Kadirkamanathan, and P. J. Fleming, "sparego - a hybrid optimization algorithm for expensive uncertain multi-objective optimization problems," in *Evolutionary Multi-Criterion Optimization* (K. Deb, E. Goodman, C. A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed, eds.), (Cham), pp. 424–438, Springer International Publishing, 2019.

[46] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 74–88, Feb 2019.

[47] A. Wurr and J. Anderson, "Multi-agent trail making for stigmergic navigation," in *Advances in Artificial Intelligence* (A. Y. Tawfik and S. D. Goodwin, eds.), (Berlin, Heidelberg), pp. 422–428, Springer Berlin Heidelberg, 2004.

[48] F. Arvin, A. E. Turgut, F. Bazyari, K. B. Arikan, N. Bellotto, and S. Yue, "Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method," *Adaptive Behavior*, vol. 22, no. 3, pp. 189–206, 2014.

[49] S. Kernbach, D. Hbe, O. Kernbach, R. Thenius, G. Radspieler, T. Kimura, and T. Schmickl, "Adaptive collective decision-making in limited robot swarms without communication," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 35–55, 2013.

[50] R. C. Purshouse, *On the Evolutionary Optimisation of Many Objectives*. PhD thesis, Department of Automatic Control and Systems Engineering. The University of Sheffield, Sheffield, UK, September 2003.

[51] K. Miranda, S. Zapotecas-Martínez, A. López-Jaimes, and A. García-Nájera, "A comparison of bio-inspired approaches for the cluster-head selection problem in wsn," in *Advances in Nature-Inspired Computing and Applications* (S. K. Shandilya, S. Shandilya, and A. K. Nagar, eds.), (Cham), pp. 165–187, Springer International Publishing, 2019.

[52] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of Theoretical Biology*, vol. 218, no. 1, pp. 1–11, 2002.

[53] A. Kolpas, M. Busch, H. Li, I. D. Couzin, L. Petzold, and J. Moehlis, "How the spatial position of individuals affects their influence on swarms: A numerical comparison of two popular swarm dynamics models," *PLOS ONE*, vol. 8, pp. 1–10, 03 2013.

[54] M. Aguilera-Ruiz, L. Torres-Treviño, and A. Rodríguez-Liñán, "Collective motion of a swarm of simulated quadrotors using repulsion, attraction and orientation rules," in *Advances in Computational Intelligence* (G. Sidorov and O. Herrera-Alcántara, eds.), (Cham), pp. 512–521, Springer International Publishing, 2017.

[55] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics Automation Magazine*, vol. 17, pp. 56–65, Sep. 2010.

[56] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, May 2011.

[57] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[58] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms: Second Edition*. Luniver Press, 2010.

[59] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 256–279, June 2004.

[60] M. Reyes-Sierra and C. Coello A. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, pp. 287–308, 01 2006.

[61] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, pp. 1945–1950, July 1999.

[62] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, April 2002.

[63] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 284–302, April 2009.

[64] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712–731, Dec 2007.

[65] N. Riquelme, C. Von Lcken, and B. Baran, "Performance metrics in multi-objective optimization," in *2015 Latin American Computing Conference (CLEI)*, pp. 1–11, Oct 2015.

[66] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 257–271, Nov 1999.

[67] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 117–132, April 2003.

[68] J. Knowles, "Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 50–66, Feb 2006.

[69] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evolutionary Computation*, vol. 26, no. 3, pp. 411–440, 2018. PMID: 29786458.

[70] J. D. Knowles, *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, Department of Computer Science. The University of Reading, Reading, UK, January 2002.

[71] T. Hesterberg, S. Monaghan, D. S Moore, A. Clipson, R. Epstein, W. H Freeman, and C. New York, "Bootstrap methods and permutation tests," *Introduction to the Practice of Statistics*, vol. 14, 01 2005.

[72] J. E. Gentle, *Computational Statistics*. Springer Publishing Company, Incorporated, 1st ed., 2009.

[73] A. Krishnamoorthy and D. Menon, "Matrix inversion using cholesky decomposition," *2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pp. 70–72, 2013.

[74] A. P. Piotrowski, M. J. Napiorkowski, J. J. Napiorkowski, and P. M. Rowinski, "Swarm intelligence and evolutionary algorithms: Performance versus speed," *Information Sciences*, vol. 384, pp. 34–85, 2017.

Autobiografía

Luis Ángel Márquez Vega

Candidato para obtener el grado de
Maestría en Ciencias de la Ingeniería Eléctrica

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

Performance evaluation and optimization of swarms of robots in a specific task

Nacido en Reynosa, Tamaulipas, México el 9 de Febrero de 1994, hijo del Ing. Ladislao Márquez Hernández y la Sra. Aracely Vega López. Egresado de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León con el grado de Ingeniero en Mecatrónica en Junio de 2017. Trabajó como practicante en el departamento de control en Bosch Rexroth de Mayo de 2016 a Abril de 2017.

Autobiography

Luis Ángel Márquez Vega

Candidate to obtain the degree of
Master of Science in Electrical Engineering

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Thesis:

Performance evaluation and optimization of swarms of robots in a specific task

Born in Reynosa, Tamaulipas, Mexico on February 9, 1994, son of Eng. Ladislao Márquez Hernández and Mrs. Aracely Vega López. Graduated from the Facultad de Ingeniería Mecánica y Eléctrica of the Universidad Autónoma de Nuevo León with the grade of Engineer in Mechatronics in June 2017. Work as intern in the control department in Bosch Rexroth from May 2016 to April 2017.