

Optimisation multi-objectif de l'ordonnancement de la collaboration homme robot dans un poste de travail

MLIK Salwa^{1,2} SAHNOUN M'hammed ¹ BENALI Abderraouf ²

1: Lineact CESI, Campus de Rouen

2: Université de Versailles-Saint-Quentin-en-Yvelines, France

salwa.mlik@ens.uvsq.fr

msahnoun@cesi.fr

abderraouf.benali@uvsq.fr

Résumé

Le contexte de travail développé au laboratoire de recherche LINEACT CESI est l'industrie de futur. L'objectif est d'optimiser la collaboration Homme-Robot dans un poste de travail collaboratif, dont l'optimisation sera sur d'allocation des ressources ainsi l'ordonnancement des tâches. Vu la complexité de ce système et l'impact multiple de l'ordonnancement du poste collaboratif, nous avons opté pour une optimisation multi-objectifs, où nous avons plusieurs fonctions objectives à minimiser parmi une liste des indicateurs de performance d'une opération collaborative, telle que : le temps, le coût, l'énergie consommée par le robot et la fatigue des opérateurs. Il existe dans la littérature un certain nombre de méthodes de résolution des problèmes d'optimisation, mais peu se sont intéressées au problème de la diversité. L'utilisation des algorithmes génétiques est une des plus prometteuses en termes de diversité. Durant ce travail, nous avons utilisé deux approches, premièrement l'algorithme génétique classique avec une fonction objective qui présente la somme pondérée des fonctions objective normalisées, la deuxième est une approche multi-objectifs, basées aussi sur l'algorithme évolutionnaire GA -algorithme génétique-, qui est la NSGA II - Non-dominated Sorting Genetic Algorithm- pour la génération de nouvelles solutions candidates, plus diversifiés. Les résultats obtenus ont montré l'efficacité des algorithmes proposés et l'importance de l'optimisation des opérations

Mots clés: Système de production collaboratif, Ordonnancement des tâches, Allocation des ressources, optimisation multi-objectives, les algorithmes génétique GA , Non-dominated Sorting Genetic Algorithm NSGA II.

1. Introduction

De nos jours, les entreprises sont en train de vivre une nouvelle révolution industrielle pour faire face à la concurrence et la complexité des exigences du marché. Elles doivent alors s'adapter rapidement aux nouvelles exigences et savoir saisir les opportunités et être compétitives. En effet, pour qu'une entreprise garde sa réputation dans le marché, elle doit assurer la qualité de ses produits, innover d'une manière continue, et avoir un système de production rapide pour répondre aux besoins variables des clients. Tout cela avec un coût optimal en gardant la marge du bénéfice. La compétitivité est associée à la notion de productivité. Il faut impérativement améliorer les équipements et les technologies de la production dans le but d'augmenter sa cadence. La collaboration homme-robot est devenue une technologie clé pour l'usine du futur. Ce système de production du futur permet de garantir la meilleure exploitation des capacités des opérateurs humains afin d'assurer les réponses les plus efficaces et les plus résilientes à un environnement complexe et dynamique [1]. La collaboration homme robot combine les points forts du travailleur humain ; à savoir son intelligence, sa dextérité, son pouvoir de la réception, le traitement des informations et la prise de décision et celles du robot assistant telle que la vitesse, l'endurance, la force. Cette collaboration permet également la mise en œuvre d'un degré variable d'automatisation des postes de travail, afin de gagner en rendement, qualité et efficacité du processus de production.

Le but de ce travail est de proposer une méthode d'optimisation multi objective de l'ordonnancement des opérations dans une tâche collaborative entre un humain et un robot. Nous avons considéré plusieurs fonctions objectives en lien avec la productivité et d'autres en lien avec l'ergonomie et le confort de l'opérateur.

Ce papier est organisé comme suit : Nous allons commencer par un état de l'art sur les différentes approches de distribution des tâches entre l'homme et le robot dans le processus d'assemblage, ainsi les facteurs de performance d'une opération collaborative. Ensuite, nous allons voir les différentes méthodes pour résoudre les problèmes d'optimisation. Dans la deuxième partie, nous allons voir le développement pour la résolution de notre problématique et les résultats obtenus avec les critères choisis. Le papier se termine par une conclusion et des perspectives de ce travail.

2. État de l'art

2.1. Approches de distribution des tâches dans le processus d'assemblage

L'assemblage est un processus de fabrication qui exige une flexibilité en termes de capacité et d'aptitude, l'être humain est la ressource la plus optimale pour cette flexibilité. Le défi est donc la collaboration entre l'humain et le robot dans ce processus pour atteindre plusieurs objectifs par cette coopération (efficacité opérationnelle, faible coût de production unitaire, l'augmentation du taux de production). Les robots sont toujours inférieurs aux humains dans plusieurs compétences, mais ils peuvent effectuer des tâches mieux que l'homme, donc nous devons répartir les tâches entre eux de façon efficace pour avoir des résultats optimaux. Alors, il est nécessaire de disposer une méthode pour évaluer les tâches selon des critères pour pouvoir donner cette décision d'affectation de la tâche au robot ou à l'homme.

Dans [2] les auteurs ont développé une méthodologie structurée pour définir le potentiel d'automatisation HRC de chaque tâche dans une opération d'assemblage pour ensuite avoir la capacité d'allocation des tâches entre l'homme et le robot. Ils ont évalué ce potentiel selon plusieurs facteurs, généralement en 3 aspects : Les propriétés physiques de composant - La caractérisation de la tâche (processus) - L'espace de travail collaboratif. (Figure 1)

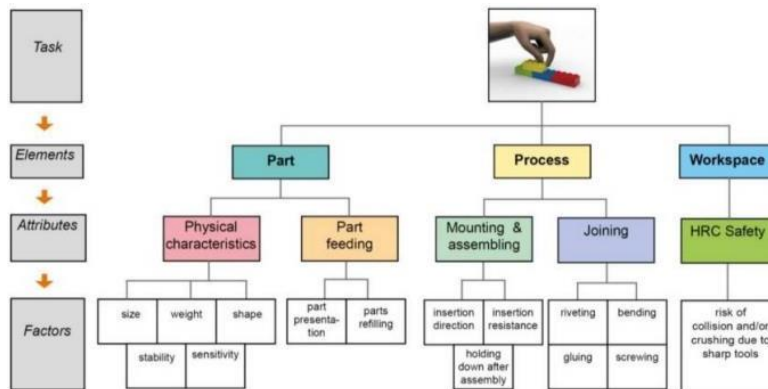


Figure 1: Structure de la méthode

Pour [3], ils ont identifié le potentiel de la collaboration homme-robot, pour une chaîne de montage de véhicules automobiles, en passant par 2 étapes : au premier lieu définir le potentiel d'automatisation de chaque mouvement dans l'ensemble de l'assemblage : Pick, Place, Move...

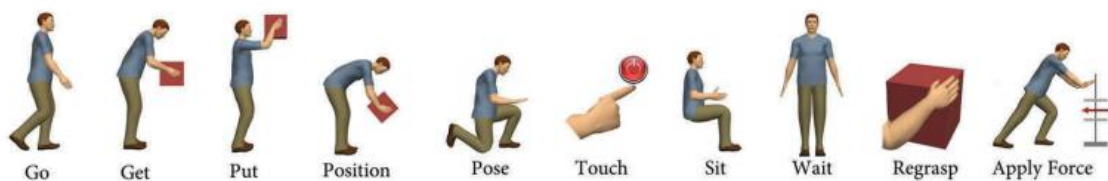


Figure 2: Quelques mouvements du processus [6]

La deuxième étape consiste à calculer le score pour chaque mouvement en considérant la répétition et la durée de ce mouvement. Cette approche se ressemble à d'autres approches qui se base sur la décomposition des tâches en sous tâches pour les décrire et les analyser afin de décider l'allocation mais se différencie en terme des critères de décision par exemple, dans [4] après la décortication hiérarchique des tâches, et pour déterminer la solution optimale de la planification de la collaboration ils ont opté à faire une analyse qualitative, où les exigences de performance : productivité (durée de l'assemblage), la qualité (erreur d'assemblage), la fatigue et la sécurité de l'opérateur humain, aussi la proximité du robot durant la collaboration qui engendre un stress chez l'opérateur. Alors que dans [5] ils ont pris comme facteur de décision les besoins de la tâche en ce qui concerne la présence du poids, la nécessité d'un déplacement, et les exigences de précision ou de dextérité, sans prendre en compte l'équilibre de la charge du travail entre les deux collaborateurs où on privilège toujours le robot s'il est capable d'accomplir la tâche comme l'homme. Il y a aussi qui ont vu qu'il faut aussi considérer l'exposition du travailleur dans la collaboration [6] ce qui veut dire les conditions ambiantes, la posture du corps et le régime horaire.

2.2. Les facteurs de performance d'une opération collaborative

Pour pouvoir évaluer la performance d'une opération collaborative, il faut se baser sur des critères qui seront les facteurs de décision de la planification de la collaboration par la suite. Dans la littérature, il existe plusieurs critères, ceux qui sont liés à la productivité et d'autre liées à l'ergonomie.

A. Critères de productivité :

Pour avoir un processus de production collaboratif performant, il faut mettre une ou plusieurs fonctions objectives soit à minimiser ou à maximiser. Dans [8] [9] [10] pour le processus d'assemblage la minimisation du temps globale (Makespan) est une mesure clé de la performance pour la productivité ainsi [9] [10] [11] le coût de la production (Calcul du coût horaire des robots, y compris le coût de l'équipement, de l'électricité et de la maintenance, paiements pour le travailleur humain par heure...) Considère un critère important pour juger la performance qu'il faut le minimiser, par exemple en minimisant [9] le temps d'attente humain ou d'inactivité en raison du coût élevé des ressources humaines, et maximiser son temps d'occupation. Aussi, pour les équipements, nous devons viser à avoir un niveau de saturation le plus élevée possible, c'est-à-dire le pourcentage du temps où le robot est utilisé [12].

B. Critères d'ergonomie :

Les critères de santé et de sécurité au travail sont d'une importance cruciale dans la mise en œuvre de la robotique collaborative. Alors il faut mettre dans les objectifs aussi des facteurs liés à l'humain non seulement la productivité. Plusieurs recherches s'intéressent au traitement de ce sujet, dans la littérature, il existe nombreux indices et méthodes pour but d'étudier l'ergonomie physique et cognitive du travailleur humain. Dans [9], pour garantir la continuité des opérations et réduire la charge psychologique, il faut planifier les tâches d'une manière fluide, cela veut dire il faut effectuer des tâches qui sont homogènes plutôt que des tâches hétérogènes, ce qui permet du travailleur humain de gagner en temps. Dans [8], le confort humain peut s'assurer en minimisant les actions répétitives, les postures contraignantes, les forces élevées, l'exposition aux vibrations des outils et des équipements. Ainsi, il faut impérativement minimiser la fatigue qui peut entraîner une baisse des performances telles qu'un ralentissement de la vitesse de réaction, une incapacité à réagir aux changements et à se concentrer, le travailleur humain alors commettra des erreurs et prendra certaines actions inconsciemment, ce qui prolonge le temps pour accomplir la tâche. Pour évaluer la fatigue, dans [12] les auteurs ont présenté une modélisation qui peut mesurer le niveau de fatigue physique de l'opérateur humain en fonction du temps dont il dispose pour exécuter les opérations. Il y a ainsi des chercheurs qui ont donné plus d'importance à la posture du travailleur et la criticité de la tâche en termes de charge musculosquelettique

pour concevoir la transformation d'un poste de travail manuel à un poste collaboratif pour un processus d'assemblage [13] en utilisant la méthode RULA - Rapid Upper Limb Assessment - qui considère les positions articulaires des membres supérieurs, du cou et du tronc (les parties du corps recrutées dans les tâches d'assemblage). Cette analyse RULA pour chaque tâche est utile pour identifier les tâches manuelles les moins adaptées du point de vue de la posture qui sont les meilleures candidates pour être potentiellement exécutées par le cobot. Garg [14] [15] a développé un indice nommé Strain Index, pour évaluer le niveau de risque d'une tâche manuelle sur l'extrémité supérieure (la main, le poignet, l'avant-bras ou le coude). Le SI est mesuré en évaluant six paramètres sur une échelle de 1 à 5 : Intensité de l'effort (I E), Durée de l'effort (DE), Efforts par minute (E M), Posture de la main et du poignet (HWP), Vitesse de travail (SW) et Durée par jour (DD), on calcule le SI en prenant le produit des multiplicateurs pour chaque paramètre selon le tableau, grâce au résultat nous pouvons classer le risque de la tâche (Faible, Moyen ou Élevé).

Pour l'évaluation de la charge mentale, [16] il existe 3 types de mesure : Mesures de performance : Le nombre d'erreurs ; Mesures physiologiques : le rythme cardiaque ou les saccades oculaires ; Mesures subjectives : à partir de questionnaires tel que : Modified Cooper-Harper scale (MCH, Wierwille & Casali, 1983), Subjective Workload Assessment Technique (SWAT, Reid & Nygren, 1988), National Aeronautics and Space Administration Task Load Index (NASA-TLX, Hart & Staveland, 1988).

Etude	Critère Objectif							Existence d'un Modèle mathématique
	Productivité		Ergonomie					
	Coût	Temps*	Fatigue	Strain index	Charge mentale	Posture	Force de manipulation	
Annika Raatza,2020 [17]		X						
George [12] Michalos,2018	X	X	X				X	X
Moreteza Dianatfar,2019[18]		X						
Ming Zhang, 2022 [19]		X	X					X
Luca Gualtieri, 2020 [20]		X						
T. Arai, 2010[21]					X			
Dalle Mura, Gino Dini, 2019 [11]	X							X
Ya-Jun Zhang, 2021[8]		X		X				
Ponda, 2010 [22]	X	X						X
Ana Colim, 2021[13]						X		
Luca Gualtieri, Ilaria Palomba,2020 [23]		X					X	
Fei Chen,2014 [9]	X	X						X
Margaret Pearce, Bilge Mutlu, 2018 [15]		X		X				X

*Temps (temps de saturation, manutention, globale, d'attente et inactivité ...)

2.3. Méthodes de résolution des problèmes d'optimisation

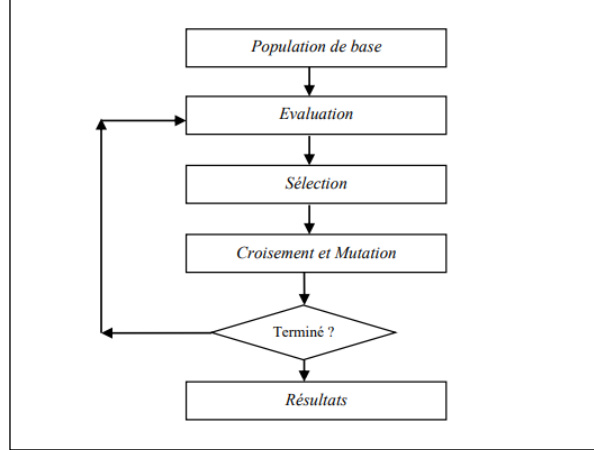
L'optimisation de la planification a fait l'objet d'une attention considérable dans le domaine de la recherche opérationnelle. En conséquence, une pléthore de méthodes et d'algorithmes ont été développés pour traiter des scénarios ou des problèmes spécifiques. Plusieurs méthodes et des algorithmes qui ont été développés pour la résolution de problèmes d'optimisation. Ces algorithmes peuvent être classés en deux méthodes : Exactes, et approchées (heuristique et métaheuristique). Les méthodes exactes permettent de trouver exactement le meilleur résultat. Mais, par exemple, les planificateurs d'assemblage qui utilisent ces méthodes qui garantissent la recherche de l'optimum global, il se trouve qu'elles sont lentes et limitées en raison de la nature du problème et sa complexité lorsque le nombre de pièces du produit à assembler augmente et cela devient plus coûteux. Pour surmonter cette complexité, les algorithmes méta-heuristiques sont développées. Ces méthodes sont plus efficaces, et capables d'échapper aux optimums locaux. Voici quelques-unes de ces méthodes qui ont été appliquées à la planification de la séquence d'assemblage dans la littérature au cours des dernières années : recuit simulé (SA) qui donne une solution unique et algorithme génétique (GA) qui donne une population de solution.

A. Recuit simulé SA :

Cette méthode de recherche a été proposée par des chercheurs d'IBM (Bonomi [24]). On utilise un processus métallurgique qui est le recuit pour trouver un minimum. En effet, pour qu'un métal retrouve une structure proche du cristal parfait, on porte celui-ci à une température élevée, et on laisse refroidir lentement de manière à ce que les atomes aient le temps de s'ordonner régulièrement. Ce processus a été transposé à l'optimisation et a donné une méthode simple et efficace, il se base sur le concept de recherche locale (Notion de voisinage) en explorant l'espace de recherche point par point commençant par une solution initiale. Dans [25] Les auteurs ont utilisé cette méthode comme résoudre les problèmes de localisation des dépôts et des centres de distribution dans une chaîne logistique Cette méthode du Recuit Simulé est facile à implémenter, elle évite les pièges des optimum locaux et donne d'excellents résultats pour un nombre de problèmes complexes, cependant, nombreux tests sont y nécessaires pour trouver les bons paramètres.

B. Algorithme génétique GA :

Les Algorithmes génétiques (AG) sont des Algorithmes d'optimisation stochastique inspirés de la génétique classique Goldberg [26]. On considère une "population initiale" qui contient des solutions initiales choisies arbitrairement. A chaque individu on associe une performance d'évaluation. Sur la base de ces performances, on classe cette population initiale selon notre objectif d'optimisation et on crée une nouvelle population de solutions en utilisant les opérateurs évolutionnaires qui sont la sélection, le croisement et la mutation. Quelques individus se reproduisent, d'autres disparaissent et seuls les individus les mieux adaptés sont supposés survivre. On recommence le cycle jusqu'à l'obtention d'une solution satisfaisante ou jusqu'à un critère d'arrêt. Dans [27] les auteurs ont utilisé GA Algorithmes génétiques pour obtenir un ensemble de solutions optimales pour leur problématique d'optimisation. Cette méthode a été ensuite comparée avec la résolution par la méthode du recuit simulé. Cette comparaison a démontré que GA dépasse SA du point de vue nombre moyen et qualité des solutions optimales.



3. Développement

La recherche de la planification d'un processus d'assemblage optimisée, avec une manière intéressante qui résout le problème de la détermination de l'ordonnancement des séquences ainsi que le problème de l'allocation des ressources vu que on est dans un poste de travail collaborative, tout en respectant les relations de précédence et en optimisant les performances, est le défi de ce travail, développer un algorithme qui permet de donner ce résultat avec une liste des fonctions objectives considérées, nous avons utilisé une méthode méta-heuristique très réputée dans la résolution des problèmes d'ordonnancement, qui est l'algorithme génétique. Mais puisque nous avons plus qu'un critère d'évaluation nous avons aussi implémenté une approche évolutionnaire, qui nous permet de prendre plusieurs critères pour évaluer la solution, cette approche est nommée la NSGA II « Non-dominated Sorting Genetic Algorithm II » proposé par Deb et al. [28] c'est un algorithme génétique multi-objectif originellement conçu afin de déterminer un optimum de Pareto, répondant à de multiples objectifs. Il s'articule autour d'une mécanique assez classique d'algorithme génétique GA.

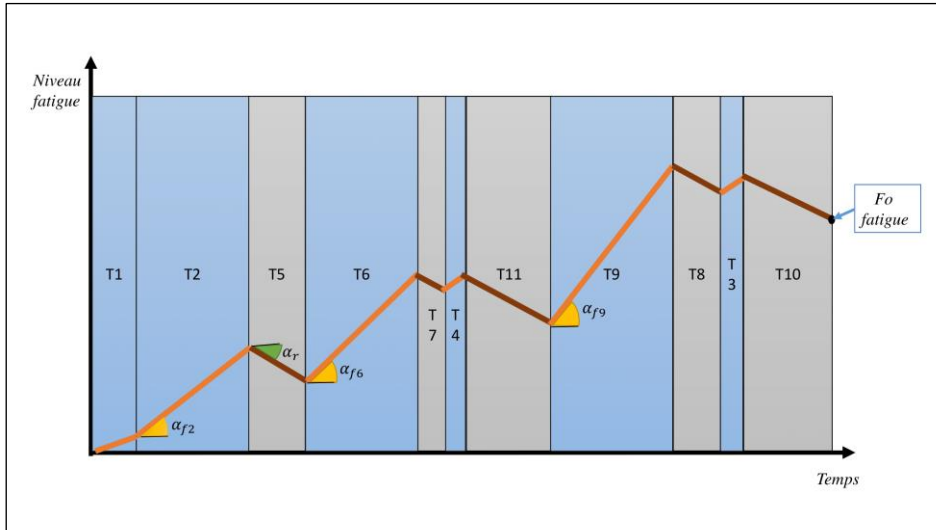
3.1. Les fonctions objectives

Dans ce travail, nous avons choisi deux fonctions objectives à minimiser liées à la productivité, Ils sont considérés dans plusieurs études qui traitent la problématique d'ordonnancement dans la littérature, la première fonction est la somme C_j où C_j est la date de fin d'exécution de la tâche T_j (Completion time), il présente le temps de début de la tâche plus son temps opératoire [29] [30], la deuxième fonction est le coût, donc le but sera la minimisation du somme C_j et le coût de production unitaire.

$$f_{o_1} = \min(\text{coût})$$

$$f_{o_2} = \min\left(\sum_{j=1}^{j=nt} C_j\right)$$

Pour la partie ergonomique du poste de travail nous avons choisis la fatigue comme fonction objective, nous avons proposé une modélisation simple pour évaluer la fatigue engendrée par un ordonnancement des tâches donnée, c'est une optimisation par une fonction qui calcule la fatigue à la fin de l'opération. En effet la fatigue augmente pendant l'exécution la tâche par l'homme avec une pente α_f qui présente le coefficient de fatigue associé à la tâche et diminue pendant le repos (temps d'attente) par un coefficient de repos α_r liées à l'opérateur.



Coefficient de la fatigue	Cf(tâche)
Coefficient du repos	Cr(opérateur)

$$Ft(0)=0$$

$$\text{Si la ressource est H ou HR : } ft_{+1} = ft_{-1} + (pt_j * cf_j)$$

$$\text{Si la ressource est R : } ft_{+1} = ft_{-1} - (pt_j * cr_{op})$$

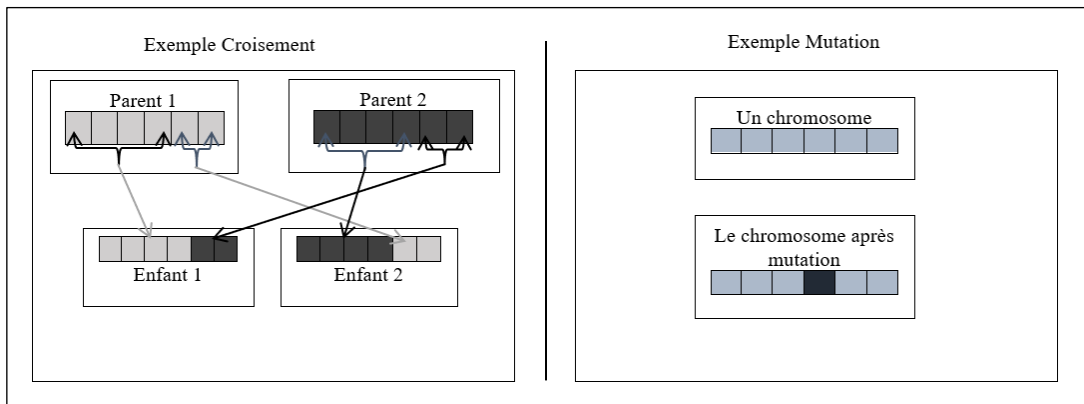
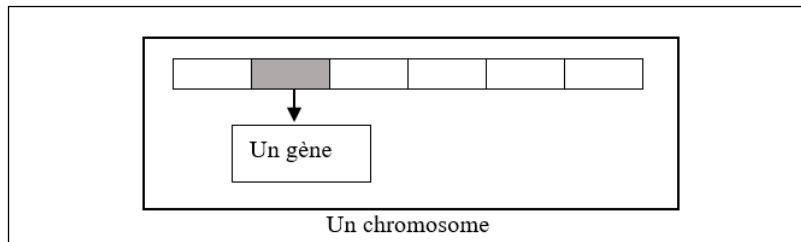
3.2. Algorithme Génétique Classique

Afin d'explorer une partie de l'ensemble des solutions possibles et de déterminer une solution la plus adaptée, l'algorithme commence par une population initiale choisie aléatoirement contient des individus « Chromosomes » qui présente un ensemble de solution possible, cette population sera évaluée selon nos objectifs définis, afin de créer une nouvelle population les individus les plus forts, ceux qui possédant le meilleur score à la fonction objectif sont sélectionnés pour intégrer la nouvelle génération, ensuite on crée des nouveaux individus par des opérations de croisement et mutation. Les croisements permettent de conserver les gènes permettant d'obtenir de bons scores chez les parents, tandis que les mutations permettent d'explorer l'espace des solutions mais également d'éviter une convergence trop rapide en explorant des solutions plus « lointaines », et maintiennent ainsi une certaine diversité. Plusieurs générations sont ainsi créées, jusqu'à un critère d'arrêt - un nombre d'itérations- (algorithme 1). La population finale sera alors composée d'individus ayant de meilleurs scores à la fonction objectif que les individus initiaux.

Algorithme 1 : Principe général de fonctionnement des algorithmes génétiques

$P \leftarrow$ Population initiale ;
 $P_c \leftarrow$ Classement de la population selon fonctions objectives
Tant que le critère d'arrêt n'est pas atteint faire
 $P_c \leftarrow$ sélection des bons individus pour les intégrer directement dans la nouvelle population ;
 $P_c \leftarrow$ croisements et mutations de P_c pour le remplissage de la nouvelle population $P+1$;
 $P+1 \leftarrow$ évaluer et classer ;
 $P_c \leftarrow P_c \cup P+1$;
Fin

- ✓ La taille de la population reste fixe dans toutes les générations
- ✓ Un croisement de chromosomes est une fonction qui permet d'obtenir de nouveaux chromosomes composés des gènes des chromosomes d'entrée (parents).
- ✓ Une mutation est une fonction qui permet d'obtenir un nouveau chromosome en modifiant un ou plusieurs gènes d'un chromosome déjà existant.



A. Résolution de la problématique de planification d'un poste collaboratif par l'algorithme génétique

a. Codage de solutions sous la forme de chromosomes

La population initiale contient un nombre de chromosome qui présente une solution aléatoire valide et exécutable pour notre problématique. Notre chromosome présente alors un ordonnancement d'un nombre défini des tâches en respectant toutes les contraintes existantes, à savoir : les relations de précédence et la répétition, et chaque tâche a une ressource associée, soit l'homme, le robot, ou l'homme et robot au même temps.

Exemple d'un chromosome de notre population initiale

T6	T4	T1	T4	T8	...	T1	T11
R	H	H	HR	R	...	HR	H

b. Evaluation du Chromosome

Chaque tâche contient des informations qui vont nous permettre d'évaluer le chromosome selon les critères de performance que nous avons choisis : la somme des dates de fin de la tâche et le coût.

Tâche	Temps opératoire(ressource)	Coût (ressource)
Si ressource = H	Pt (H)	Coût (H)
Si ressource = R	Pt (R)	Coût (R)
Si ressource = RH	Pt (HR)	Coût (HR)

Alors comme indiqué dans l'exemple de chromosome, selon la ressource associée à la tâche, nous allons récupérer les données dont nous aurons besoin pour calculer le score du chromosome.

- ✓ Temps de la disponibilité de la ressource pour calculer C_j la date de fin de la tâche :

Dans notre problématique, nous avons 2 types de ressources qui génère 3 possibilités d'allocation : l'homme, le robot, ou l'homme et robot travaillent ensemble sur la même tâche. Alors il y'a certainement des tâches qui peuvent commencer en parallèle et des tâches qui nécessite la disponibilité du robot et de l'homme pour s'exécuter. Cela va influencer principalement notre critère d'optimisation qui offre la possibilité de comparer les individus : la somme C_j . C'est pour cela nous avons introduit la notion disponibilité de la ressource pour pouvoir évaluer correctement notre chromosome.

Rappelant que C_j c'est le temps de début de la tâche plus son temps opératoire, le temps de début de la tâche sera le temps de disponibilité de ressource. C'est-à-dire, si la tâche va être effectuée par une seule ressource, l'homme ou le robot, le temps de début de la tâche c'est égal au temps de disponibilité de la ressource pour l'exécuter. Mais si la tâche sera effectuée par l'homme et le robot ensemble, le temps de début est le maximum entre les deux temps de disponibilité de ressources.

c. Génération de la nouvelle population

i. Sélection

Dans ce travail nous avons utilisé la sélection par rang, Cette étape de la sélection permet d'identifier les individus susceptibles d'être croisés dans une population. Il existe plusieurs techniques de sélection [31]. Nous présentons les quatre les plus utilisées parmi elles :

- ✓ Sélection par rang : consiste à ranger les individus de la population dans un ordre croissant ou décroissant, selon l'objectif (fonction fitness).
- ✓ Sélection par roulette : elle consiste à créer une roue de loterie biaisée pour laquelle chaque individu de la population occupe une sélection de la roue proportionnelle à sa valeur d'évaluation.
- ✓ Sélection aléatoire : cette sélection se fait aléatoirement, uniformément. Chaque individu a donc une probabilité uniforme d'être sélectionné.

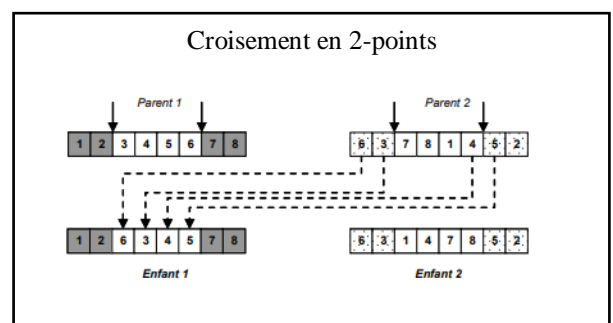
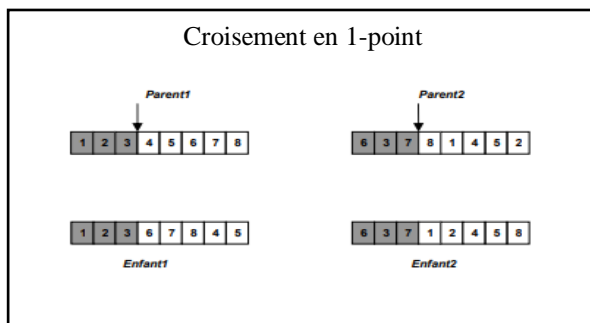
- ✓ Sélection par tournoi : le tournoi le plus simple consiste à choisir aléatoirement un nombre k d'individus dans la population et à sélectionner celui qui a la meilleure performance. Les individus qui participent à un tournoi sont remis ou sont retirés de la population, selon le choix de l'utilisateur.

ii. Opération de croisement

L'opération de croisement permet d'enrichir la population en manipulant les gènes des chromosomes. Un croisement est envisagé avec deux parents et génère un ou deux enfants. Selon la littérature [32], plusieurs opérateurs de croisement sont proposés, parmi les plus utilisés :

Croisement en 1-point : consiste à diviser chacun des deux parents en deux parties à la même position, choisie au hasard et à recopier la partie inférieure du parent à l'enfant et à compléter les gènes manquants de l'enfant à partir de l'autre parent en maintenant l'ordre des gènes. C'est le type de croisement que nous avons utilisé pour la génération des nouveaux individus, il nous permet d'éviter de tomber dans la problématique de répétition des tâches.

Croisement en 2-points : Ce type de croisement est utilisé en choisissant aléatoirement 2 points de coupure pour dissocier chaque parent en 3 parties. Ceux en extrémités pour le Parent1 (respectivement Parent2) sont copiés à l'Enfant1 (respectivement Enfant2). On complète la partie restante de l'Enfant1 par les éléments du Parent2 et la partie restante de l'Enfant2 par les éléments du Parent1 en balayant de gauche à droite et en ne reprenant que les éléments non encore transmis.



iii. Opération de mutation

La mutation est nécessaire pour une exploration efficace de l'espace. Elle permet de quitter les optimums locaux. Comme pour les croisements, de nombreuses méthodes de mutation ont été présentées dans la littérature [32]. Nous citons les plus connues :

- Opérateur d'inversion simple : cet opérateur consiste à choisir aléatoirement deux points de coupure et inverser les positions des gènes situés au milieu ; c'est le type de mutation que nous avons exploité.
- Opérateur d'insertion : cet opérateur consiste à sélectionner au hasard un gène et une position dans le chromosome à muter, puis à insérer le gène sélectionné dans la position choisie.

d. Solution avec fonction Somme pondérée

Dans notre problématique nous avons plus qu'une fonction fitness pour évaluer le chromosome, alors c'est un problème multi-objectif. Pour pouvoir comparer entre le chromosome avec l'algorithme génétique classique nous avons évalué par une fonction somme pondérée des fonctions objectives. La première étape

sera est la normalisation des données, cela veut dire qui la mise à l'échelle les données de manière à ce qu'elles soient bornées entre [0,1]. Le maximum sera égal à 1 et le minimum sera égal à 0. Cela par soustraire, pour chaque variable, la valeur minimum et à diviser le résultat par l'écart maximum rencontré.

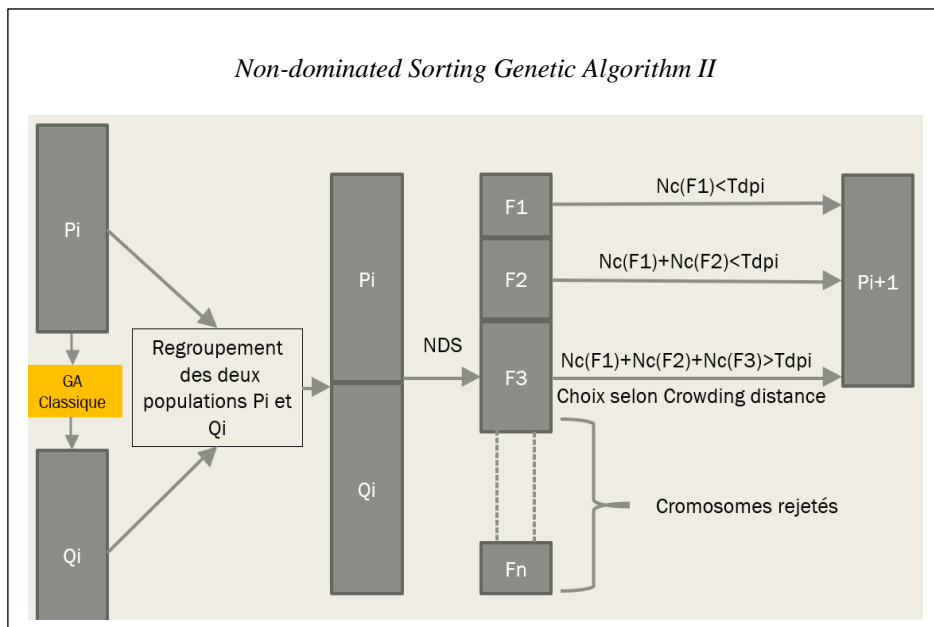
Alors après avoir calculer les valeurs des critères d'optimisation pour chaque individu, on peut avoir la valeur min et max de chaque fonction objectives dans la population. Ensuite nous pouvons normaliser toutes les valeurs des critères d'évaluation pour qu'on puisse les injecter dans la fonction somme pondérée.

$$f_{op} = \sum_{i=0}^{i=nf} w_i f_{o_{n_i}}$$

nf est le nombre de fonction objective

3.3. Non-dominated Sorting Genetic Algorithm II

Après avoir développé l'algorithme génétique, nous avons implémenté l'algorithme NSGA II qui se base sur GA classique. La figure suivante présente l'algorithme de NSGA II



Après La génération de la population Q_i par l'AG classique, les deux populations (P_i et Q_i) sont regroupées et ordonnées en différents fronts (les individus non dominés sont dans le front 1, ceux dominés uniquement par ceux du front 1 sont dans le front 2, ...). L'étape suivante c'est le remplissage de la population P_{i+1} , Les fronts seront sélectionnés par ordre croissant jusqu'à ce qu'un front ne puisse pas intégrer entièrement la population. Ce front est alors trié selon la Crowding distance et seuls les meilleurs individus en fonction de ce score sont sélectionnés. La principale nouveauté de cet algorithme NSGA II est l'intégration de ce processus de conservation des distances : Crowding distance

L'objectif de la Crowding Distance est de préserver la diversité dans la population. Pour cela, elle va affecter à chaque individu un rang en fonction de la proximité de son score par rapport aux individus ayant les scores les plus proches.

- ✓ Plus un individu est éloigné des autres, meilleur est son classement.

Cette distance est calculée en fonction des scores des individus pour les fonctions objectives. Pour chaque fonction objective, les chromosomes sont triés et leurs scores de distances sont calculés à partir de la différence qui les séparent de leurs voisins les plus proches.

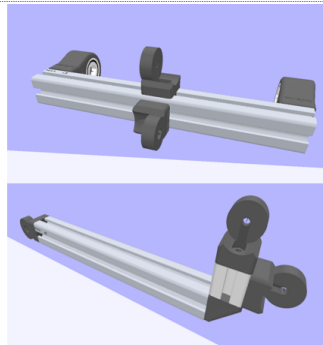
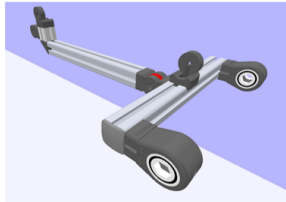
La distance finale d'un individu est la somme de ses distances pour chaque fonction objective.

$$CD_{C_i} = \sum_{i=1}^{NBf_o} \frac{f_{o_i}(C_{i+1}) - f_{o_i}(C_i - 1)}{f_{o_i} \max - f_{o_i} \min}$$

4. Résultats

4.1. Use case

Dans cette étude, nous allons appliquer notre solution de planification sur un poste de travail collaboratif, un bras manipulateur type UR10 (Universal Robot) et l'opérateur humain. L'objectif sera, planifier le montage d'un produit (Kit INFENTO, vélo pour enfants). Plus particulièrement le poste 1 où s'effectue l'assemblage du cadre inférieur du vélo. Ce cadre est divisé en 2 sous-ensembles (figure.).



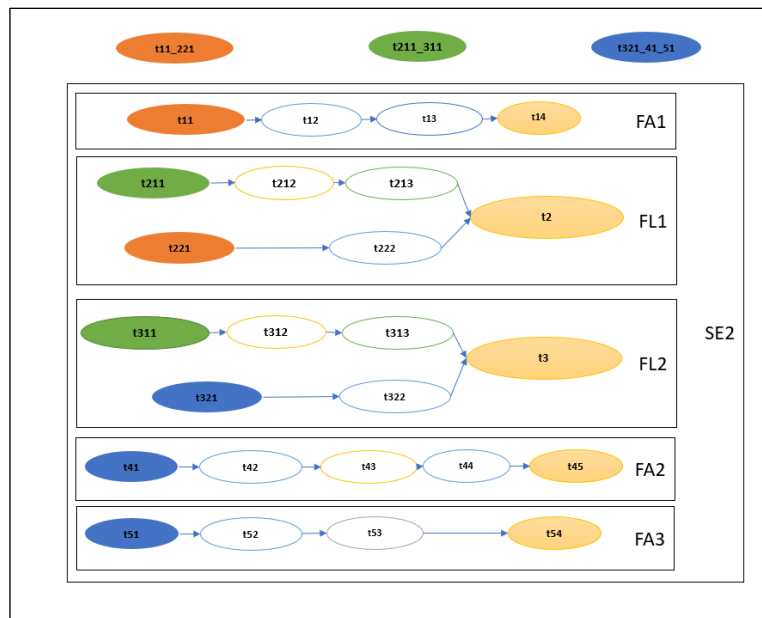
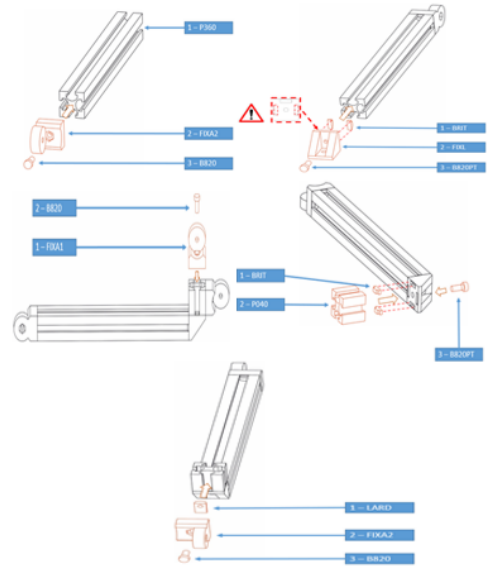
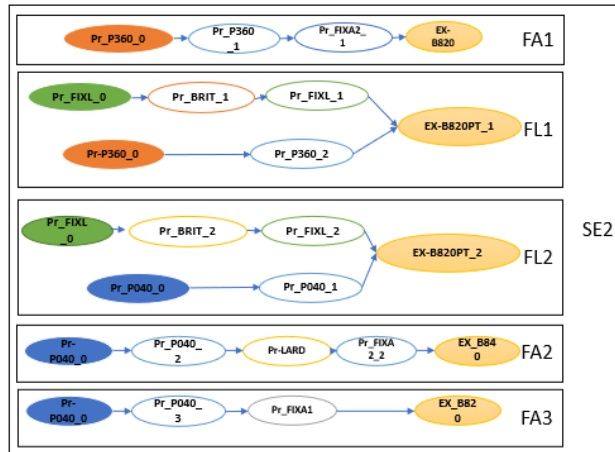
Part 1

Part 2

Nous allons appliquer nos algorithmes sur la planification de l'assemblage du sous ensemble 2, premièrement nous avons étudié les relations de précedence pour extraire les séquences interdites.

- *Sous ensemble 2*

Ces schémas résument les relations de précedence entre les tâches et leurs codages.



4.2. Résultats

Pour la planification de l'assemblage du sous ensemble 2 du cadre inférieur, nous avons implémenté dans le code les données d'entrées (input) pour les données on a supposé que le taux horaire du profil d'un assembleur puisse atteindre plus de 12 euros bruts, et que le taux horaire du Robot UR10 est : le coût fix d'achat + le coût de maintenance + le coût de la consommation.

Pour la configuration de l'algorithme, on a pris une taille de population de 100 chromosomes et un critère d'arrêt de 50 itérations, nous avons obtenue des résultats avec l'algorithme génétique classique puis avec la NSGA2, les deux méthodes sont performantes sauf que la NSGA2 nous permette d'éviter la pondération des fonctions objectives.

Pour pouvoir illustrer la performance d'optimisation de nos deux approches avec nos critères choisis, nous allons comparer entre une solution d'ordonnancement et allocation de la population initiale, et une solution de la population finale.

a. Solution avec l'algorithme génétique classique

Pour avoir les résultats avec l'algorithme génétique classique, sachant que nous avons plusieurs critères à optimiser, nous les a pondérés sous une fonction somme pondérée avec le même poids.

1. Solution de la population initiale

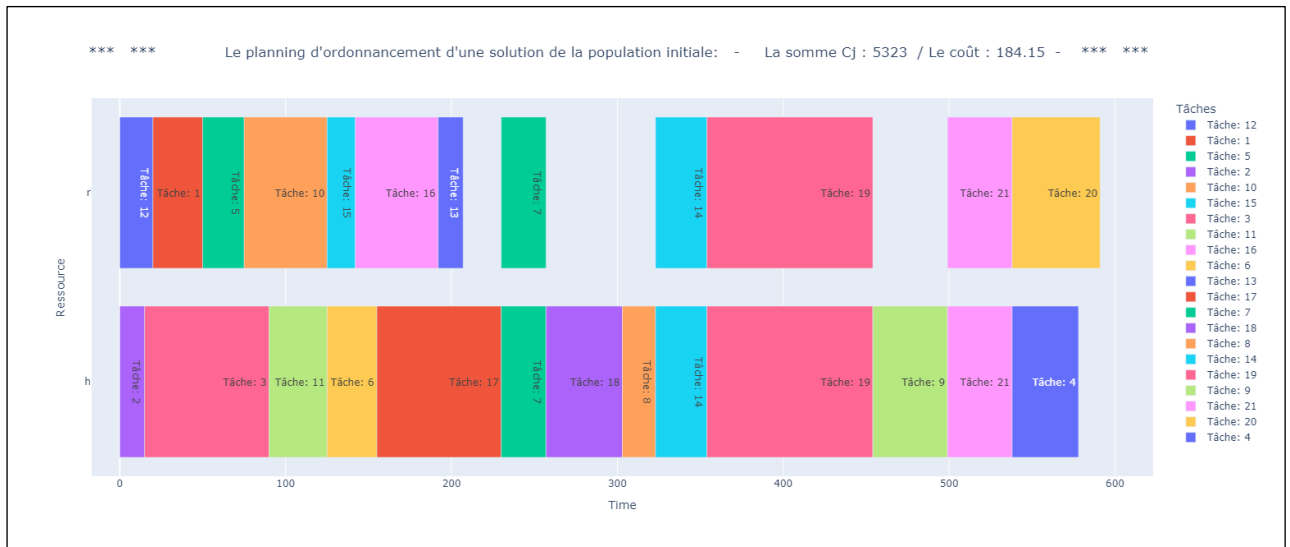


2. Solution de la population finale



b. Solution avec l'algorithme NSGA2

1. Solution de la population initiale



2. Solution de la population finale



Grâce au code que nous avons développé sur python nous sommes capables de planifier les tâches et les distribuer entre l'homme et le robot de façon optimisée. Alors pour pouvoir l'utiliser pour planifier un poste de travail donné, il faut tout d'abord étudier les contraintes à savoir : la répétition des tâches, et les relations de précédence entre les tâches. Aussi il faut collecter les données de temps opératoire, le coût, la fatigue pour chaque tâche. Il y a ainsi une flexibilité pour ajouter un autre critère d'optimisation, le code était structuré pour cet objectif, avec une programmation orientée objet [33].

Dans l'application de notre cas d'usage les données ont été estimées pour tester le bon fonctionnement de l'algorithme, les résultats ont montré qu'il donne des résultats satisfaisants, la prochaine étape sera la mesure exacte des données pour avoir une planification du poste de travail collaboratif et la programmer dans le robot UR10 du poste de l'assemblage de cadre inférieur du vélo et créer un mode opératoire pour les opérateurs. Ensuite utiliser le même code pour planifier tous les postes collaboratifs du montage du vélo en manipulant les données d'entrée, à savoir : le nombre des tâches, les données des critères d'optimisation, et les séquences interdites.

5. Conclusion et Perspectives

Dans ce travail, nous avons développé une solution pour l'optimisation multi-objectif des postes de travail collaboratif, l'ordonnancement des tâches avec l'allocation des ressources humain (H), robot (R) et humain robot (HR) en utilisant une méthode d'optimisation basée sur l'algorithme génétique multi-objectif NSGA II. Les objectifs d'optimisation considérés sont le temps, le coût et la fatigue. Nous l'avons appliqué sur un poste de travail de montage du cadre inférieur d'un vélo pour enfant, les résultats obtenus par les deux méthodes, l'algorithme génétique classique et la NSGA II sont sous forme de planning du processus entier avec la distribution des tâches, ils sont satisfaisants et répondent parfaitement aux contraintes de précédence. Comme perspective de ce travail, nous comptons développer encore plus notre solution, et s'intéresser plus particulièrement à l'optimisation de l'ergonomie du poste de travail collaboratif. L'idée est de développer une solution qui permet de faire le lien entre la fatigue et la localisation de la tâche. Cela va permettre d'intégrer le côté de charge mentale de l'opérateur dans le processus d'optimisation. Par la suite,

l'intelligence du robot va permettre de guider l'opérateur, prédire ses intentions et savoir ses futurs gestes. Ceci est très utile pour améliorer la sécurité et pour augmenter la qualité de la collaboration entre le robot et l'opérateur.

Remerciement

Ce travail a été développé dans le cadre de projet de fin d'étude pour la validation du Master de recherche, Robotique, assistance et mobilité, au sein du laboratoire Lineact CESI Campus de Rouen. A l'occasion de ce stage, j'ai eu à rencontrer des personnes aimables dont rien ne pourra m'enlever les instants que nous avons partagés ensemble. C'est pour cette raison, pour terminer ce rapport je tiens à exprimer mes sincères remerciements aux différentes personnes sans lesquels ce projet n'aurait pas eu lieu ou n'aurait pas abouti. Que toute l'équipe de recherche trouvent ici mes sincères remerciements pour la collaboration et leur disponibilité à fournir les informations. Je ne peux terminer cette section sans témoigner toute ma gratitude et ma reconnaissance à toute l'équipe pédagogique de Master RAM pour leurs efforts pour nous transmettre les diverses compétences que nous avons aujourd'hui et les opportunités qu'ils nous ont offert.

Références

- [1] Maddi Kunta, P. K. R., Pham, Q. V., Prabadevi, B., Deepa, N., Dev, K., Gadekallu, T. R., ... & Liyanage, M. (2021). Industry 5.0: A survey on enabling technologies and potential applications. *Journal of Industrial Information Integration*, 100257.
- [2] Malik, A. A., & Bilberg, A. (2019). Complexity-based task allocation in human-robot collaborative assembly. *Industrial Robot: the international journal of robotics research and application*.
- [3] J Teiwes, T. Bänziger, A. Kunz and K. Wegener, "Identifying the potential of human-robot collaboration in automotive assembly lines using a standardised work description," 2016 22nd International Conference on Automation and Computing (ICAC), 2016, pp. 78-83, doi: 10.1109/IconAC.2016.7604898
- [4] Tan, J. T. C., Duan, F., Kato, R., Arai, T., & Hall, E. (2010). *Collaboration planning by task analysis in human-robot collaborative manufacturing system*. INTECH Open Access Publisher.
- [5] Antonelli, D., & Bruno, G. (2017). Dynamic task sharing strategy for adaptive human-robot collaborative workcell. *DEStech Transactions on Engineering and Technology Research*, (icpr).
- [6] Müller, R., Vette, M., & Mailahn, O. (2016). Process-oriented task assignment for assembly processes with human-robot interaction. *Procedia CIRP*, 44, 210-215.
- [7] Malik, A. A., Masood, T., & Bilberg, A. (2020). Virtual reality in manufacturing: immersive and collaborative artificial-reality in design of human-robot workspace. *International Journal of Computer Integrated Manufacturing*, 33(1), 22-37.
- [8] Zhang, Y. J., Liu, L., Huang, N., Radwin, R., & Li, J. (2021). From manual operation to collaborative robot assembly: an integrated model of productivity and ergonomic performance. *IEEE Robotics and Automation Letters*, 6(2), 895-902.
- [9] Chen, F., Sekiyama, K., Cannella, F., & Fukuda, T. (2013). Optimal subtask allocation for human and robot collaboration within hybrid assembly system. *IEEE Transactions on Automation Science and Engineering*, 11(4), 1065-1075.
- [10] Ponda, S., Choi, H. L., & How, J. (2010). Predictive planning for heterogeneous human-robot teams. In *AIAA Infotech@ Aerospace 2010* (p. 3349).
- [11] Dalle Mura, M., & Dini, G. (2019). Designing assembly lines with humans and collaborative robots: A genetic approach. *CIRP Annals*, 68(1), 1-4.
- [12] Michalos, G., Spiliotopoulos, J., Makris, S., & Chrysosouris, G. (2018). A method for planning human robot shared tasks. *CIRP journal of manufacturing science and technology*, 22, 76-90.
- [13] Colim, A., Faria, C., Cunha, J., Oliveira, J., Sousa, N., & Rocha, L. A. (2021). Physical ergonomic improvement and safe design of an assembly workstation through collaborative robotics. *Safety*, 7(1), 14.
- [14] J. Steven Moore & Arun Garg (1995) The Strain Index: A Proposed Method to Analyze Jobs for Risk of Distal Upper Extremity Disorders, *American Industrial Hygiene Association Journal*, 56:5, 443-458
- [15] Pearce, M., Mutlu, B., Shah, J., & Radwin, R. (2018). Optimizing makespan and ergonomics in integrating collaborative robots into manufacturing processes. *IEEE transactions on automation science and engineering*, 15(4), 1772-1784.
- [16] Cegarra, J., & Morgado, N. (2009, September). Étude des propriétés de la version francophone du NASATLX. In *EPIQUE 2009: 5ème Colloque de Psychologie Ergonomique* (pp. 233-239). France : Nice.
- [17] Raatz, A., Blankemeyer, S., Recker, T., Pischke, D., & Nyhuis, P. (2020). Task scheduling method for HRC workplaces based on capabilities and execution time assumptions for robots. *CIRP Annals*, 69(1), 13-16.
- [18] Dianatfar, M., Latokartano, J., & Lanz, M. (2019). Task balancing between human and robot in mid-heavy assembly tasks. *Procedia CIRP*, 81, 157-161.
- [19] Zhang, M., Li, C., Shang, Y., & Liu, Z. (2022). Cycle Time and Human Fatigue Minimization for Human-Robot Collaborative Assembly Cell. *IEEE Robotics and Automation Letters*, 7(3), 6147-6154.
- [20] Gualtieri, L., Rauch, E., & Vidoni, R. (2021). Methodology for the definition of the optimal assembly cycle and calculation of the optimized assembly cycle time in human-robot collaborative assembly. *The International Journal of Advanced Manufacturing Technology*, 113(7), 2369-2384.

- [21] Arai, T., Kato, R., & Fujita, M. (2010). Assessment of operator stress induced by robot collaboration in assembly. *CIRP annals*, 59(1), 5-8.
- [22] Ponda, Sameera S., Han-Lim Choi, and Jonathan P. How. "Predictive Planning for Heterogeneous Human-Robot Teams." *AIAA Infotech@Aerospace 2010* 20-22 April 2010, Atlanta, Georgia, AIAA 2010-3349.
- [23] Gualtieri, L., Palomba, I., Merati, F. A., Rauch, E., & Vidoni, R. (2020). Design of human-centered collaborative assembly workstations for the improvement of operators' physical ergonomics and production efficiency: A case study. *Sustainability*, 12(9), 3606.
- [24] E. Bonomi, J.-L. Lutton, Le recuit simulé, Pour la Science, numéro 129, pages 68-77, juillet 1988. 5.4.
- [25] Anthony Ross, Vaidyanathan Jayaraman *Computers & Industrial Engineering* 55 (2008) 64–79.
- [26] Kanchan Das (2011) A quality integrated strategic level global supply chain model, *International Journal of Production Research*, 49:1, 5-31.
- [27] Fulya Altiparmak , Mitsuo Gen , Lin Lin , Turan Paksoy *Computers & Industrial Engineering* 51 (2006) 196–215.
- [28] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2) :182–197, A
- [29] Vijaya Chandru; Chung-Yee Lee; Reha Uzsoy (1993). *Minimizing total completion time on a batch processing machine with job families.*, 13(2), 61–65. doi:10.1016/0167-6377(93)90030-k.
- [30] Sun Lee, Ik (2018). Minimizing Total Completion Time in the Assembly Scheduling Problem. *Computers & Industrial Engineering*, (), S0360835218302730–. doi: 10.1016/j.cie.2018.06.001
- [31] Dréo, J., A. Péreowski, P. Siarry et E. Taillard, 2003. Métaheuristiques pour l'optimisation difficile. Editions Eyrolles.
- [32] Nearchou, A. C., 2004. The effect of various operators on the genetic search for large scheduling problems. *International Journal of Production Economics*, 88 :191–203.
- [33] https://github.com/SalwaMLIK/nsg2_salwa