THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Dmitrii ARKHIPOV

Le 23 septembre 2019

Planification socio-responsable du travail dans les chaînes de montage d'aéronefs : comment satisfaire à la fois objectifs ergonomiques et économiques

---

Ecole doctorale : **AA - Aéronautique, Astronautique**

Spécialité : **Génie Industriel**

Unité de recherche :

Thèse dirigée par
Olga BATTAÏA et Julien CEGARRA

Jury

M. Alexandre Dolgui, Rapporteur
M. Jacques Carlier, Rapporteur
M. Fabio Sgarbossa, Examinateur
Mme Daria Battini, Examinatrice
M. Cyril Briand, Examinateur
M. Alexander Lazarev, Examinateur
Mme Olga BATTAÏA, Directrice de thèse
M. Julien Cegarra, Co-directeur de thèse

Socio-responsible planning of work in aircraft assembly lines: how to satisfy both ergonomic and economic objectives

Contents

Résumé français

Introduction

Contexte général

La concurrence mondiale féroce oblige les entreprises manufacturières à rechercher l'efficacité opérationnelle. La planification et l'utilisation des ressources sont l'un des facteurs clés de l'agenda des responsables de production qui nécessitent une optimisation constante. Un effort scientifique considérable est consacré au développement d'algorithmes et de politiques de contrôle intelligents et fiables basés sur l'analyse des données produites et enregistrées dans le contexte de l'Industrie 4.0, où de plus en plus de machines et de robots intelligents sont déployés [1, 2]. Cependant, la mise en œuvre d'une telle approche dans des environnements de fabrication centrés sur l'homme où les ressources humaines sont fortement impliquées dans le processus de fabrication n'est pas simple. Dans cette situation, même le mot "optimisation" peut être interprété négativement dans certains contextes.

Pour des raisons évidentes, dans la pratique, les facteurs humains ne peuvent être ignorés lorsqu'il est question de planifier des activités de fabrication impliquant des opérateurs humains. Ils jouent même un rôle décisif dans la fiabilité et l'efficacité de la planification réalisée et comme conséquence des systèmes de fabrication. Cependant, les procédures de planification existantes ignorent généralement ces facteurs humains souvent considérés comme difficiles à évaluer avec précision [3]. Non seulement les données sont plus difficiles à enregistrer, mais le stockage et le traitement de telles données hautement personnalisées posent de nombreux défis en matière de protection de la vie privée et l'application de réglementations en matière de protection des données, telles que le règlement européen GDPR (règlement général sur la protection des données en Europe) [4].

En même temps, le défi de l'amélioration des conditions de travail a récemment été mis en avant dans les 17 objectifs du développement durable adoptés par les Nations Unies en 2015 (objectif 8 "Travail décent et croissance économique" et objectif 12 "Consommation et production responsables"). Selon les statistiques disponibles, en 2017, 2,78 millions de personnes sont décédées des suites d'une activité professionnelle, contre 2,33 millions selon les estimations de 2011 [5, 6]. Les coûts mondiaux et européens des accidents du travail et des maladies professionnelles sont considérables : le coût mondial s'élève à 2 680 milliards d'euros, soit 3,9 % du PIB mondial et le coût européen à 476 milliards d'EUR, ce qui correspond à 3,3 % du PIB européen. [7].

Dans ce contexte, l'objectif des responsables de production est de fournir des conditions de travail sûres à leurs employés afin de préserver leur santé, d'améliorer leur bien-être et de réduire l'absentéisme. En raison de la prise de conscience croissante de l'impact des facteurs humains sur les performances finales des systèmes de fabrication, le nombre d'études sur leur intégration dans les procédures de planification augmente actuellement [8]. Cependant, l'écart est encore considérable. En particulier, les modèles développés pour l'environnement d'assemblage automobile ne conviennent pas au contexte d'assemblage d'aéronef où chaque opérateur exécute une longue séquence de tâches variables tandis que, dans des périodes de courte durée, généralement utilisées dans l'assemblage automobile, les travailleurs sont principalement concernés par des tâches répétitives. Des temps de cycle plus longs offrent plus de possibilités de changer d'activité et de faire des pauses. A présent, nous allons décrire l'organisation des chaînes de montage des aéronefs plus en détail.

Organisation des lignes de montage

La fabrication des aéronefs est un processus longue et complexe qui se termine par les chaînes de montage final des aéronefs (FAL). C'est la dernière étape qui suit la fabrication de toutes les pièces et l'assemblage préliminaire de différentes composantes. Dans une FAL, l'assemblage d'un aéronef est complété, la cabine est personnalisée et les tests finaux sont réalisés. La conception de la FAL et l'organisation du processus d'assemblage ont un impact direct sur le temps d'assemblage d'un aéronef, donc sur le nombre total d'aéronefs assemblé par an, sur la qualité globale et le coût d'assemblage d'un aéronef. Une FAL représente généralement près de 30% du coût total de fabrication [9], [10].

Les FALs ont des cadences relativement faibles et emploient beaucoup de main-d'œuvre. Son organisation doit satisfaire à une variété de contraintes et d'objectifs.

Les FAL peuvent être classés en trois groupes, en fonction des méthodes de localisation et de déplacement des cellules :

- Fixes lignes d'assemblage ;
- Pulse lignes d'assemblage ;
- Mobile lignes d'assemblage.

Pulse Assembly Lines ou des lignes de montage taktées ou cadencées est l'organisation de chaînes de montage la plus utilisée dans l'industrie aéronautique. C'est ce type qui est considéré dans cette thèse.

Lignes d'assemblage cadencées

Une telle ligne d'assemblage est composée d'un ensemble de postes de travail synchrones [11]. Le travail d'assemblage total à effectuer sur la ligne est divisé en séquences de tâches à réaliser sur chaque poste de travail. Chaque aéronef traverse la ligne et se rend à tous les postes de travail installés. La durée totale de toutes les tâches affectées à un poste de travail doit être inférieure à l'intervalle de production nommé takt time. Chaque poste de travail possède tous les équipements, ressources et opérateurs nécessaires pour la réalisation des tâches à exécuter. Chaque aéronef parcourt la ligne dans l'ordre des postes de travail installés jusqu'à la fin du processus de son assemblage sur le dernier poste de travail.

Les avantages de telles lignes d'assemblage :

- Moins d'investissement requis.
- Flexibilité accrue et tolérance aux pannes.
- Logistique interne facilitée.
- Une plus grande variété du travail effectué par les opérateurs.

Leurs inconvénients :

- La planification et l'exécution des tâches doivent être supervisées en vue de respecter la contrainte de l'intervalle de production.
- Si des tâches utilisant un équipement coûteux sont affectées à des postes de travail différents, cet équipement doit être dédoublé ou son utilisation doit être synchronisée.

L'image suivante montre un exemple de chaîne d'assemblage cadencée (Fig. 1.1).

Problèmes de planification dans les chaînes de montage aéronautiques

Des prévisions optimistes pour le trafic aérien mondial futur qui se traduisent par une demande sans cesse croissante dans ce secteur industriel pèsent lourdement sur l'efficacité et la fiabilité des chaînes d'assemblage final des avions [12, 13, 14].

Pour des raisons techniques et financières, les chaînes de montage d'aéronefs se caractérisent par un nombre de postes de travail relativement restreint, un nombre considérable

Figure 1: Lockheed Martin F22 Pulse FAL. Source: Lockheed Martin.

de tâches à effectuer sur chaque poste de travail et une cadence relativement faible par rapport à d'autres secteurs industriels comme automobile. De par ces caractéristiques, elles diffèrent considérablement des chaînes de montage automobile où le nombre de postes de travail est généralement très élevé, mais où le nombre de tâches par poste de travail est limité en raison de cadence élevée. Pour cette raison, les techniques de planification développées pour l'environnement automobile ne peuvent pas être appliquées directement pour l'assemblage d'aéronefs et doivent être adaptées à cet environnement de fabrication particulier.

De plus, pendant le processus d'assemblage de l'aéronef, les opérateurs effectuent des tâches manuellement dans des postures associées à des risques ergonomiques élevés. Pour améliorer les performances économiques et sociales des chaînes de montage d'aéronefs, il est nécessaire de développer de nouvelles méthodes de planification tenant compte des critères à la fois économiques et ergonomiques des opérateurs. Cependant, ce n'est que récemment que les travaux de [15, 16, 17] ont présenté les premières méthodes de planification intégrant explicitement les données ergonomiques. Toutes les études citées ont été réalisées dans le contexte de la production automobile où la répétitivité des tâches est beaucoup plus élevée que dans les chaînes de montage d'aéronefs. Une analyse exhaustive des travaux portant sur l'intégration de critères ergonomiques dans les méthodes de planification des chaînes de montage a récemment été présentée dans [18]. Il en ressort que les conditions ergonomiques peuvent être améliorées sans ou avec un impact faible sur les performances économiques des chaînes de montage à court terme, tandis que les investissements réalisés génèrent des résultats très positifs à long terme.

L'objectif de cette thèse est de prendre en compte les aspects économiques et ergonomiques lors de la planification des tâches dans les chaînes de montage des aéronefs, en tenant compte des exigences suivantes :

- l'utilisation optimale des ressources, y compris du temps, de l'espace, des équipements spéciaux, et des ressources humaines ;
- la minimisation des erreurs d'opérateur ;
- la réduction du temps de cycle ;
- l'amélioration des conditions de travail.

Il doit être mentionné que, contrairement aux chaînes de montage automobile étudiées de manière approfondie, dans le contexte des chaînes de montage d'aéronefs, un nombre important d'opérateurs et un grand nombre de tâches doivent être assignés à un petit nombre de postes de travail. Le problème de la planification des tâches sur chaque poste de travail peut être considéré comme un cas particulier du problème de planification du projet avec contraintes de ressources (RCPSP). La difficulté réside dans le fait que le RCPSP a été prouvé être NP-difficile au sens fort du terme [19] même dans sa formulation classique.

La résolution du RCPSP est liée à la sélection des modes d'exécution des tâches et à l'attribution des intervalles de traitement des tâches. Dans ce problème, la disponibilité des ressources est supposée limitée ; par conséquent, les relations de précédence et les contraintes de ressources doivent être prises en compte.

Deux types de relations de précédence sont considérés. Dans la formulation du RCPSP classique, les relations de précédence établissent une séquence prédéfinie entre deux activités, la deuxième activité ne peut commencer que lorsque la première est terminée. Les relations de précédence généralisées permettent de définir des décalages horaires entre les heures de début de deux activités. La formulation du RCPSP avec décalage temporel est référencée dans la littérature comme RCPSP/max.

Un ensemble de ressources renouvelables représente des éléments tels que des équipements, des ressources humaines qui sont utilisés, mais ne sont pas consommés lors de la réalisation du projet.

De plus, pour chaque contexte particulier, des contraintes spécifiques supplémentaires peuvent être formulées. Dans cette thèse, nous portons une attention particulière aux contraintes d'ergonomie.

En effet, le processus d'assemblage d'un aéronef comprend une grande variété d'opérations manuelles imposant des postures liées à des risques ergonomiques élevés. En ergonomie, de nombreuses études ont été réalisées sur les risques professionnels, mais leurs résultats ne sont pas intégrés dans des procédures de planification. Habituellement, les entreprises évaluent les risques professionnels des unités de travail existantes (en France, conformément aux articles L.4121-2 et L.4121-3 du Code du travail). Cette évaluation est conçue pour identifier un large éventail de risques. Notre objectif est d'utiliser les méthodes de notation existantes au stade de la planification afin d'éviter des risques ergonomiques élevés. L'objectif de cette étude est d'intégrer les modèles ergonomiques à des méthodes de planification appropriées dans le contexte des chaînes de montage d'aéronefs.

Évaluation des risques ergonomiques dans les systèmes de production

L'un des principaux objectifs des directeurs de production est de fournir des conditions de travail sûres à leurs employés afin de préserver leur santé, d'améliorer leur bien-être et de réduire leur taux d'absentéisme. L'exposition à des risques sur le lieu de travail peut engendrer ou contribuer au risque de développer des maladies liées au travail, par ex. troubles musculo-squelettiques (TMS). Selon Occupational Safety and Health Administration (OSHA, USA, [20]), les facteurs suivants sont principalement responsables des TMS : force, répétition, postures contraignantes, postures statiques, mouvements rapides, compression ou stress au contact, vibrations. De plus, souvent, ces facteurs agissent en combinaison tout en augmentant considérablement la probabilité de causer un TMS [21]. Afin d'identifier les facteurs de risques professionnels, l'OSHA conseille d'évaluer les aspects suivants du travail : exigences physiques du travail, conditions et disposition des lieux de travail et des postes de travail, caractéristiques des objets manipulés ou utilisés et conditions environnementales. Les facteurs psychosociaux suivants doivent également être analysés : rythme de travail, demandes de tâches, autonomie, monotonie, soutien social, cycle travail / repos, incertitude professionnelle. Il convient également de noter que l'OSHA classe les assembleurs au 4e rang des dix professions les plus concernées par les TMS [20].

Evaluation des risques physiques

Pour évaluer les risques ergonomiques, un certain nombre de méthodes directes, de méthodes d'observation, de méthodes subjectives et d'autres méthodes psychophysiologiques ont été développées par des ergonomes. L'intégration de telles approches dans les procédures de planification ou d'ordonnancement n'est pas toujours facile en raison de la complexité de l'évaluation.

Parmi les méthodes les plus couramment utilisées, les suivantes ont déjà été associées aux problèmes de planification dans les systèmes de production de la littérature :

- NIOSH-Eq – *Equation de levage de l'Institut national de la sécurité et de la santé au travail* [22]. L'Institut national pour la sécurité et la santé au travail (NIOSH) a mis au point le Guide des pratiques de travail pour le levage manuel en 1981 afin d'aider les praticiens de la sécurité et de la santé à évaluer les risques liés à la levée et à la baisse d'emplois dans le but de prévenir et de réduire les faibles maux de dos chez les travailleurs et les blessures connexes. Ce manuel a été révisé et étendu en 1991 et fait actuellement partie de la norme ISO 11228.

- LI – *Index de levage* [23]. L'index de levage affiche le rapport entre la charge manipulée et une charge recommandée. Ce dernier dépend du poids de l'objet manipulé, des emplacements horizontaux et verticaux, de la distance, de l'angle de symétrie, de la fréquence de levage, de la durée et du couplage entre les mains et l'objet.

- RULA – *Évaluation rapide des membres supérieurs* [24]. L'outil RULA propose des feuilles de travail permettant une évaluation rapide des risques ergonomiques liés aux membres supérieurs, au cou et au tronc. Il consiste en une feuille de calcul et ne nécessite que sept étapes pour calculer le score final. Le score final dépend des forces appliquées, des postures inconfortables et statiques et de la fréquence de répétition.

- REBA – *Évaluation rapide de tout le corps* [25]. La REBA est une méthode spécialement conçue pour être sensible au type de postures de travail imprévisibles rencontrées dans les secteurs de la santé et des autres services. Elle a été développée par une équipe d'ergonomes, de physiothérapeutes, d'ergothérapeutes et d'infirmières, qui ont rassemblé et codé individuellement plus de 600 exemples posturaux.

- OCRA – Outil relatif aux actions répétitives sur les professions [26]. L'indice OCRA évalue les manipulations répétitives à haute fréquence effectuées par les membres supérieurs et est calculé séparément pour chaque main. L'indice final d'OCRA est calculé comme le rapport entre la fréquence réelle et la fréquence recommandée des actions, mesurée en nombre de répétitions par minute. La fréquence recommandée des actions dépend des forces appliquées, des postures et de facteurs de risque supplémentaires, tels que les vibrations.

- JSI – *Indice de contrainte du travail* [27]. La JSI utilise une méthodologie similaire à celle d'OCRA avec deux paramètres supplémentaires : la vitesse de travail et la durée de la contrainte.

- DND – *Dosage quotidien du bruit* [28]. Le MDN établit la limite d'exposition recommandée pour le bruit au travail, en fonction de la période d'affection, du volume sonore et de la fréquence.

- EAWS – *Fiche de travail de l'évaluation ergonomique* [29]. L'outil EAWS évalue les postures, les forces d'action, la manipulation manuelle des matériaux, ainsi que d'autres facteurs de risque pour le corps entier et les charges répétitives des membres supérieurs. Les résultats de l'estimation de l'EAWS sont deux valeurs de risque globales : les points de risque pour l'ensemble du corps et les points de risque pour les membres supérieurs. Les points de risque plus élevés indiquent des risques plus élevés de troubles musculo-squelettiques.

- EnerExp – *méthode de dépense d'énergie* [30]. Méthode de dépense d'énergie basée sur la quantité d'énergie dépensée par un opérateur lors de l'exécution de l'activité.

Nous pouvons conclure que les méthodes d'évaluation ergonomiques existantes permettent d'estimer l'impact négatif du processus de travail sur la santé de l'opérateur en mettant l'accent sur la prévention d'un ensemble de maladies et de blessures. Certaines méthodes prennent en compte les paramètres individuels de l'opérateur, tels que l'âge, le sexe, la taille, le poids.

Problème de planification de projet avec des ressources limitées

Le problème de planification de projet à ressources limitées (RCPSP) est un problème classique de la théorie d'ordonnancement. Ce problème est connu pour être NP-difficile au sens fort [19]. Il existe de nombreuses formulations et des approches de résolution différentes pour trouver des solutions optimales et sous-optimales pour le RCPSP. Une analyse exhaustive de la littérature a été présentée dans [31]. Plusieurs comparaisons d'algorithmes existants ont été réalisées dans la littérature [32], [33], [34], [35], [36].

La formulation du RCPSP de base

La formulation de base du RCPSP peut être donnée comme suit. Un ensemble de tâches N doit être réalisé dans l'horizon de planification T à l'aide d'un ensemble de ressources renouvelables R. La capacité de la ressource $r \in R$ est égale à la valeur constante c_r. Pour toute tâche $j \in N$, les paramètres suivants sont donnés :

- p_j – temps de traitement ;

- a_{jX} – montant requis de la ressource $X \in R$;
- r_j – heure de disponibilité, l'heure à laquelle la tâche j peut être lancée au plus tôt ;
- D_j – date limite, la dernière heure pour terminer la tâche j.

Les relations de précédece entre les tâches sont données par le graphe acyclique dirigé $G(N, E)$. Si un arc $(i \to j) \in E$ existe, cela signifie que la tâche i doit être terminée avant l'heure de début de la tâche j.

Une solution π est faisable pour l'ensemble des ressources R et des tâches N, si pour tout $j \in N$ *l'heure de début* $S_j(\pi) \geq r_j$ est définie et toutes les contraintes de précédence et de capacité des ressources sont satisfaites. $\Pi(N, R)$ note l'ensemble des solutions réalisables. L'objectif est de trouver une solution réalisable avec le makespan minimal (la durée du projet la plus courte), c'est-à-dire

$$\min_{\pi \in \Pi(N,R)} \max_{j \in N} C_j(\pi), \tag{1}$$

où $C_j(\pi) = S_j(\pi) + p_j \leq D_j$ – le *temps d'achèvement* de la tâche j, où D_j – la date limite de la tâche j.

Programmation linéaire en nombres entiers mixtes

L'une des approches les plus populaires pour résoudre le RCPSP est la programmation linéaire mixte à nombres entiers (MILP). Il existe de nombreux modèles de MILP pour le RCPSP, dont la plupart sont mentionnés dans les tudes bibliographiques suivantes [37], [38]. Les formulations de MILP sont généralement classées en trois catégories conformément à la modélisation du partage du temps et des ressources. L'existence de bornes inférieures efficaces permet d'accélérer la résolution du problème et d'obtenir des solutions de meilleure qualité avec des temps de résolution plus courts.

Bornes inférieures du RCPSP

Dans la littérature, un certain nombre d'algorithmes permettant de calculer les bornes inférieures pour le makespan ont été proposés. Les instances de benchmark PSPLIB [39] est généralement utilisé pour comparer les performances des algorithmes de résolution et des bornes inférieures. Les comparaisons récentes peuvent être trouvées dans les études suivantes : [40] et [41]. L'analyse de la complexité de certains algorithmes et de la qualité des bornes obtenues a été présentée dans [42]. Cependant, les algorithmes qui calculent les bornes inférieures calculables rapidement ne fournissent généralement pas les meilleures estimations alors que les méthodes permettant d'obtenir de meilleures bornes reposent principalement sur la coopération entre la programmation linéaire et la programmation par contraintes et peuvent nécessiter un effort de calcul considérable. Dans cette thèse, nous proposons Un nouvel algorithme polynomial pour trouver une borne inférieure de makespan pour RCPSP avec des capacités de ressources variables dans le temps. Son idée est basée sur une évaluation consécutive de paires de ressources et de leur charge de travail cumulée. En utilisant l'algorithme proposé, plusieurs bornes pour les instances de benchmark PSPLIB ont été améliorées. Les résultats pour les applications industrielles sont également présentés, où l'algorithme pourrait fournir de bornes efficaces, même pour de très grandes instances. Ces contributions ont été présentées dans les publications suivantes : textit D. Arkhipov, O. Battaïa, A. Lazarev. Un algorithme pseudo-polynomial efficace pour trouver une limite inférieure sur le makespan du problème de planification de projet à ressources limitées, European Journal of Operational Research, 275 (1). 35-44 ; *D. Arkhipov, O. Battaïa, A. Lazarev. Un nouvel algorithme polynomial pour calculer les limites supérieures de l'utilisation des ressources en cas de problème RCPSP. 16ème Conférence internationale sur la gestion de projet et la planification, 17-20 avril, Rome, Italie, 2018.*

Cet algorithme de calcul de bornes peut être appliqué non seulement à la formulation classique du RCPSP, mais également à de généralisation du RCPSP. La complexité de l'algorithme est $O(r^2n^2(n+m))$ opérations, où n – nombre de tâches, r – nombre de ressources, m – nombre de points d'arrêt de fonctions de capacité de ressources et T est la longueur de l'horizon.

Expériences numériques

L'algorithme a été implémenté en C++. Deux séries d'expériences numériques ont été réalisées avec un processeur Intel Core i7 à 2,8 GHz avec 16 Go de RAM. Dans le premier, l'algorithme a été testé sur les instances de benchmark PSPLIB [39]. Dans le second cas, l'algorithme a été appliqué à des instances RCPSP de grande taille basées sur des données réelles fournies par Kuznetsov Design Bureau. Les résultats des tests sont présentés dans les tableaux 4.1 et 4.2, respectivement.

Pour la première série de tests la borne fournie par notre algorithme n'est pas pire que la meilleure connue pour 66% des instances, l'écart moyen étant d'environ 2 % . De plus, la valeur de la borne inférieure a été améliorée pour 4 instances. Il convient de noter que le temps de calcul était très court.

La deuxième série de tests a été réalisée sur des données industrielles réelles afin d'évaluer la possibilité d'appliquer l'approche présentée à des instances réelles de grande taille. Notre algorithme a été capable de trouver des bornes inférieures efficaces en temps de calcul relativement court.

Programmation par contraintes pour résoudre le RCPSP

Il existe une vaste gamme d'algorithmes de programmation de contraintes pour trouver une solution optimale/sous-optimale pour RCPSP ou pour faciliter la résolution du problème. La programmation par contraintes est largement utilisée pour résoudre ce problème [43], [44] et [45].

Les propagateurs sont utilisés comme un outil de pré-traitement des données pour resserrer les domaines de tâches et ajouter de nouvelles relations de précédence. Dans cette thèse, nous avons proposé De nouveaux propagateurs de contraintes. Les résultats obtenus ont été présentés dans la publication suivante : *D. Arkhipov, O. Battaia, A. Lazarev, G. Tarasov, I. Tarasov. Nouveaux propagateurs de domaine de tâches à complexité polynomiale pour un problème de planification de projet avec ressources limitées. IXe Conférence internationale sur l'optimisation et les applications (OPTIMA2018), 1-5 octobre, Petrovac, Monténégro, 2018.*

Les propagateurs de recherche Edge, Extended Edge et Time Tabling ont pu être améliorés grâce à la propagation de la capacité des ressources. Un nouvel algorithme de détection des relations de précédence a été également développé.

Intégration des contraintes ergonomiques dans le RCPSP

Ce chapitre aborde le problème de l'affectation de travail des opérateurs des lignes de montage d'aéronefs (FAL) sous contraintes économiques et ergonomiques. Nous supposons que les tâches à exécuter sur chaque poste de travail ont déjà été définies et que l'ensemble des opérateurs possédant les compétences requises est déjà affecté à chaque poste de travail. Le problème d'optimisation concerne un poste de travail avec ses tâches et ses opérateurs. Le problème d'optimisation considéré a pour objectif d'attribuer toutes les tâches aux opérateurs disponibles dans le respect des contraintes économiques (takt time) et ergonomiques. Ce problème est considéré comme un cas particulier de problème de planification de projet avec contraintes de ressources (RCPSP). Un modèle de programmation par contraintes (CP) est développé pour résoudre ce problème. Les contributions de ce chapitre ont également été présentées dans les publications suivantes : *D. Arkhipov, O. Battaia, A. Lazarev, J. Cegarra. Problème d'affectation des opérateurs dans les chaînes de montage d'aéronefs : une nouvelle approche de planification tenant compte des contraintes économiques et ergonomiques. Proceedia CIRP, Volume 76, 63-66, 2018* ; *D. Arkhipov, O. Battaia, J. Cegarra, A. Lazarev. Planification du travail dans les chaînes de montage à faible volume sous contraintes ergonomiques. Proceedia CIRP, Volume 72, 786-789, 2018.*

Nous avons développé de nouveaux modèles mathématiques afin d'intégrer les contraintes ergonomiques dans le RCPSP. Ces modèles sont adaptés à la planification de tâches dans les chaînes de montage d'aéronefs et peuvent prendre en compte des paramètres tels que les compétences professionnelles, deux types de contraintes ergonomiques physiques évaluées par diverses méthodes ergonomiques, ainsi que les paramètres axés sur la cognition utilisés ici pour réduire la charge cognitive des opérateurs. La méthode de résolution est basée sur la procédure en deux étapes comprenant la programmation par contraintes et la programmation linéaire en nombres entiers.

Le modèle présenté a été testé sur deux instances de données industrielles. Des expériences ont été effectuées sur le logiciel IBM ILOG CPLEX en utilisant le processeur Intel (R) Core (TM) i5-4670 à 3,40 GHz et 16 Go de RAM.

La première instance est caractérisée par 289 tâches avec 340 présences réparties en 79 groupes, 12 ressources, 7 opérateurs avec 3 spécialités et 1 méthode d'évaluation ergonomique. Pour cette instance, une solution optimale du problème de planification des tâches a été trouvée en 15 secondes. La solution optimale du problème d'affectation d'opérateur a été trouvée en 18 secondes.

Pour la deuxième instance, Les tâches étant déjà programmées, seule l'affectation des opérateurs a été optimisée. Cette instance est caractérisée par 447 tâches réparties en 79 groupes, 5 opérateurs avec 2 spécialités et 1 méthode d'évaluation ergonomique. Dans ce cas, la solution optimale a été trouvée en 8 heures. Nous avons également considéré le cas sans contraintes d'ergonomie physique. Cette relaxation nous a permis d'utiliser certaines techniques supplémentaires pour améliorer la vitesse de recherche et obtenir une solution optimale en 36 secondes.

Une autre série d'expériences numériques a été réalisée pour vérifier la cohérence du prétraitement à l'aide de propagateurs de domaine de tâches développés. Soixante-dix instances ont été générées de manière aléatoire sur la base des données de deux instances industrielles mentionnées ci-dessus. Le modèle CP développé pour le problème de planification des tâches avec la demande agrégée a été appliqué pour résoudre cet ensemble d'instances à l'aide d'IBM CP Optimizer avec et sans pré-traitement. Les résultats montrent que l'utilisation de propagateurs développés diminue le temps de calcul jusqu'à 30 % . Les propagateurs sont particulièrement efficaces pour les instances de grande taille. Des expériences numériques ont montré que cette approche peut être utilisée pour résoudre efficacement des instances industrielles réelles, même avec un grand nombre de tâches (jusqu'à 500). Toutefois, l'approche proposée n'évaluant pas l'équité de la répartition de la charge de travail entre les opérateurs, il convient d'étudier ce facteur dans les recherches futures.

Planification des effectifs dans les chaînes de montage des aéronefs

Dans le contexte mondial caractérisé par une demande difficilement prévisible de produits personnalisés et une complexité croissante des produits, des processus et des systèmes de production, le capital humain est toujours considéré comme une source de flexibilité interne capable de résoudre des problèmes imprévisibles. Pour se protéger des perturbations, de nombreuses entreprises mettent en place des programmes de formation polyvalente afin de former des travailleurs polyvalents. La littérature indique que la formation croisée peut améliorer la satisfaction, la confiance et la motivation des employés [46]. Naturellement, l'organisation des programmes d'apprentissage s'accompagne d'investissements et de temps et, comme la rétention des compétences est conditionnée par la pratique, les responsables ont pour objectif d'évaluer le besoin en opérateurs polyvalents.

Nous avons proposé un modèle mathématique pour résoudre ce problème. Lors de la première étape, l'objectif est de définir le programme de formation polyvalente afin de déterminer le nombre optimal de opérateurs polyvalents. À la deuxième étape, lorsque la composition de l'équipe est connue, l'objectif est de programmer un grand nombre de tâches tout en optimisant les conditions de travail des opérateurs.

Un modèle de programmation par contraintes a été créé pour le problème considéré. L'approche de planification développée a été appliquée à un ensemble de données d'instances de problèmes basé sur des données industrielles et a montré des résultats prometteurs en termes de temps de résolution et d'objectifs atteints. Des expériences numériques ont montré que le modèle développé permet de trouver des solutions optimales et sous-optimales d'instances de grande taille en un temps raisonnable. De plus, les procédures de pré-traitement basées sur les propagateurs de contraintes développés dans cette thèse ont été utilisées pour améliorer l'efficacité de la recherche et ont fourni des résultats prometteurs.

Le modèle présenté a été implémenté avec IBM CP Optimizer, processeur Intel (R) Core (TM) i5-4670 à 3,40 GHz, 16 Go de RAM. Les données expérimentales ont été générées sur la base de cas industriels collectés. Le jeu de données total comprend 85 instances avec un nombre de tâches compris entre 10 et 80, un nombre de relations de précédence allant de 10 à 2841, 5 spécialités, 10 zones, des heures de prise entre 500 et 8000 unités de temps et à partir de 10 à 20 opérateurs disponibles à l'embauche. Le nombre de relations de précédence augmentant avec le nombre de tâches, davantage d'instances ont été générées pour un plus grand nombre de tâches.

Deux séries d'expériences ont été réalisées. Dans la première série, les instances ont été résolues par le modèle implémenté dans IBM CP Optimizer. Dans la deuxième série, les domaines de tâches et les relations de précédence ont été propagés par des procédures de programmation de contraintes développées dans cette thèse avant la phase de solution. Au total, 41 instances ont été résolues en 30 minutes. L'utilisation de propagateurs développés pour ces instances a permis d'économiser 51 minutes, ce qui représente 13,5 % du temps consacré à la résolution des 41 instances non propagées. Pour les 44 autres cas, l'écart entre la fonction objectif obtenue et la limite inférieure évaluée était inférieur à 10,3 % de la valeur objectif.

Conclusion

Remarques finales

La prise en compte des facteurs humains dans la planification et la planification des tâches dans les chaînes de montage des aéronefs peut contribuer à améliorer la gestion des risques économiques et ergonomiques. À cette fin, les méthodes ergonomiques existantes doivent être intégrées aux modèles mathématiques utilisés dans la planification.

Dans cette thèse, nous avons considéré en particulier le cas des lignes d'assemblage aéronautiques dans le but d'intégrer les facteurs ergonomiques dans la planification des tâches d'assemblage. Il a été démontré que ce problème pouvait être modélisé comme un cas particulier de problème de planification de projet avec contraintes de ressources, en termes de problème de planification, mais il n'a jamais été étudié sous des contraintes d'ergonomie dans la littérature. Dans un premier temps, nous avons analysé les méthodes ergonomiques existantes pour l'évaluation de la charge de travail physique et celles déjà utilisées dans la planification d'applications dans les systèmes de production.

Un nouvel algorithme pseudo-polynomial permettant de calculer une limite inférieure de makespan pour RCPSP a été présenté. Cette approche est basée sur l'évaluation de l'utilisation la plus élevée possible d'une ressource par rapport à une autre ressource. Nous avons également montré comment cet algorithme peut être adapté pour une formulation du RCPSP généralisée avec une fonction de capacité constante par morceaux et une durée continue. Il a été montré que pour cet algorithme, l'ensemble des points d'arrêt peut être utilisé à la place des créneaux horaires. Cela permet de réduire la complexité de l'algorithme en polynôme $O(r^2n^2(n+m))$ nombre d'opérations, où n – nombre de tâches, r – nombre de ressources, m – nombre de points d'arrêt dans la fonction de capacité des ressources.

Nous avons proposé des nouvelles techniques de propagation de la fonction de capacité des ressources et des extensions des propagateurs existants basés sur les ressources (recherche de bord, recherche de bord étendue, affichage dans le temps) à une formulation du RCPSP généralisée avec une fonction de capacité constante par morceau. Certaines techniques de renforcement ont été suggérées pour ces propagateurs. Il a été démontré que la limitation de la capacité des ressources peut augmenter l'efficacité de la propagation. Il convient de noter que tous ces nouveaux propagateurs peuvent être utilisés dans une instruction généralisée avec des propagateurs classiques basés sur des relations de priorité ou des relations disjonctives afin de resserrer les domaines de tâches.

Les premiers modèles intégrant des facteurs ergonomiques dans le RCPSP ont été proposés. Deux formulations de problèmes différents ont été développées sous forme de RCPSP généralisé avec des contraintes de facteurs humains intégrant des méthodes d'évaluation ergonomiques. Selon les conclusions basées sur l'activité musculaire, des contraintes ergonomiques physiques ont été définies non seulement pour l'horizon de planification (heure du takt), mais également pour des intervalles plus courts. La fonction objectif makespan a été considérée pour le premier modèle et la minimisation de la charge cognitive en tant que fonction objectif pour le second modèle orientée vers la réduction de la probabilité d'erreur de l'opérateur. L'utilisation de ces modèles pour la planification du processus d'assemblage de l'avion permet de réduire l'impact négatif du processus d'assemblage sur la santé des travailleurs. Des expériences numériques sur des données réelles ont montré la cohérence des modèles.

La planification de l'effectif a été également étudiée. Le modèle présenté prend en compte les contraintes du processus d'assemblage (y compris les facteurs humains) et aide à définir les profils d'opérateurs à recruter et à former. Le modèle de programmation de contraintes proposé a été testé avec et sans l'utilisation de propagateurs de contraintes développés dans le prétraitement des données. Les résultats des expériences numériques confirment que le modèle peut être utilisé pour résoudre efficacement le problème d'optimisation et que l'utilisation de propagateurs de contraintes peut contribuer à réduire le temps de résolution.

Perspectives d'avenir

L'objectif de cette recherche était de développer une nouvelle approche de planification capable d'améliorer les conditions de travail sur les chaînes d'assemblage final des aéronefs grâce à une répartition des tâches intelligente. L'approche de planification développée a été testée sur des études de cas industrielles et a montré des résultats prometteurs en termes de temps de solution et d'objectifs atteints. Tous les modèles suggérés peuvent servir de base au développement futur d'outils de planification pour les chaînes de montage d'aéronefs. Il reste encore beaucoup à faire car, dans la pratique, trop peu de problèmes d'optimisation sont résolus efficacement, le processus de décision est essentiellement un processus manuel. Dans cette étude, nous avons modélisé une grande variété de contraintes et de fonctions objectives liées au processus d'assemblage des aéronefs, y compris celles liées aux facteurs humains. Toutefois, pour chaque configuration de chaîne de montage particulière, des contraintes et objectifs supplémentaires peuvent être pertinents et doivent être correctement modélisés.

Les recherches ultérieures peuvent notamment être consacrées à la création de nouveaux modèles mathématiques et de méthodes de résolution du problème des interruptions imprévues (absence du travailleur, fourniture tardive de pièces ou problème de qualité), pour lesquels une solution de rééchelonnement doit être trouvée très rapidement compromettre le respect du temps tact. Comme l'approche développée vise à respecter les contraintes ergonomiques réglementaires et les restrictions individuelles, mais ne compare pas si tous les travailleurs sont facturés de la même manière en termes de temps de travail ou de charge ergonomique, l'objectif d'équité dans la répartition du travail devrait être formulé et étudié.

D'autres perspectives de recherche concernent le développement de propagateurs de contraintes, en particulier de nouveaux propagateurs polynomiaux. Il est intéressant d'approfondir l'idée de la propagation non seulement des variables de solution, mais de toutes les variables d'entrée utilisées dans les contraintes et les objectifs. De plus, des variables supplémentaires (à savoir des disjonctions, des limites supérieures d'utilisation des ressources) peuvent être utilisées et propagées pour améliorer l'efficacité du processus de planification. L'effet synergique des propagateurs doit également être exploré.

Chapter 1

Introduction

1.1 General context

Fierce global competition forces manufacturing companies to strive for their operational efficiency. One of the key factors on the agenda of production managers that constantly requires optimization is the planning and use of resources. An enormous scientific effort is dedicated for developing intelligent and reliable algorithms and control policies that are based on the analysis of the data produced and recorded in the context of Industry 4.0 where more and more smart machines and robots are deployed [1, 2]. However, the implementation of such an approach in human-centred manufacturing environments where human resources are highly involved in the manufacturing process is not straightforward. In this situation, even word optimization may be interpreted negatively in some contexts.

For evident reasons, in practice, human factors cannot be ignored when it comes to plan manufacturing activities involving human operators. They even play a decisive role in reliability and efficiency of realized planning and as a consequence of manufacturing systems. However, the existing scheduling procedures generally ignore such human factors often considered as hard to be evaluated with precision [3]. Not only the data is more difficult to record, but also the storage and processing of such highly personalized data comes with many privacy challenges and the application of data protection regulations, such as EU GDPR (Europe General Data Protection Regulation), shows the importance of this question [4].

At the same time, the public attention to the challenge of improving working conditions has been recently drawn in the 17 sustainable development goals adopted by United Nations in 2015 (Goal 8 "Decent Work and Economic Growth" and Goal 12 "Responsible Consumption and Production"). According to available statistics, in 2017, 2.78 million people died from work-related causes in comparison to 2.33 million estimated in 2011 [5, 6]. The global and European costs of work-related accidents and illnesses are considerable: the global cost at EUR 2 680 billion, which is 3.9 % of global GDP and the European cost is EUR 476 billion, which, at 3.3% of European GDP [7].

In this context, the goal of production managers is to provide safe working conditions for their employees in order to preserve their health, to improve their well-being and to reduce absenteeism. Due to the increasing awareness about the impact of human factors on the final performance of the manufacturing systems, the number of studies on their integration in planning and scheduling procedures is currently growing [8]. However, the gap is still considerable. In particular, the models developed for automotive assembly environment are not suitable for aircraft assembly context where each operator performs a long sequence of variable tasks while under short takt times usually in practice in automotive assembly, the workers are mainly concerned with task repetitively. Longer takt times offer more possibilities to change of activity and to have breaks. In the next section, we describe the organization of Aircraft Assembly Lines in more details.

1.2 Organization of assembly aircraft lines

Aircraft manufacturing is a long time process which consists of long production chains, which completes at the Aircraft Final Assembly Line (FAL). It is the last stage that comes after the manufacturing of all parts and preliminary assembly of different parts in bigger aircraft portions. At FAL, these portions are assembled, the cabin is customized and final tests of aircraft are realized. The design of FAL and organization of the assembly process has a direct impact on the aircraft volume, overall quality and assembly cost. FAL is quite expensive, it accounts for nearly 30% of the total manufacturing cost [9], [10].

The aircraft assembly is low-volume labour-intensive process which have to satisfy a variety of constraints and objectives. To fulfill a diverse customer demand, aircraft manufacturers have to be able to offer a wide range of models within various airplane class, this leads to a low-volume production.

FAL can be classified into three groups, depending on the methods of location and movement of airframes:

- Fixed Assembly Lines;
- Pulse Assembly Lines;
- Moving Assembly Lines.

Let us briefly describe all three types.

1.2.1 Fixed Assembly Line

Fixed Assembly Line (or Dockyard Assembly Line) is a conventional aircraft assembly line, at which each aircraft is fixed at one place. All required equipment and resources, such as parts, supports, tools, human resource and other materials are brought to this single location to perform all the tasks.

The major advantages of Fixed Assembly Line are [47], [48]:

- Performing jobs of various types increases the skill and satisfaction of employees.
- Very high product flexibility.
- Simplification of facilities design.
- High fault tolerance, since all aircrafts are assembling independently.

Main disadvantages of Fixed Assembly Line are:

- Higher level of inventory at all levels and hence higher inventory cost.
- Hard to adapt to the fluctuation of the market.
- Production planning is complicated.
- Increasing the volume of product requires more assembly positions.
- Operators have to move between different positions.
- Assembly quality inconsistent due to different operation groups.

This organization was is the first used but currently due to the increasing demand more and more manufacturers change to the Pulse or Moving Assembly Lines similar to lines used in automobile industry where the total work is divided among several workstations.

1.2.2 Pulse Assembly Line

Pulse Assembly Line consists of a set of synchronous workstations [11]. Total assembly work to be performed in the line is divided in sequences of tasks to be performed at each workstation. Each aircraft moves through the line visiting all installed workstations. The sequence of tasks assigned to a workstations has to respect the same for the line makespan (takt time). Each workstation possesses all the necessary equipment, resources and workers according to the sequence of tasks to be performed. All tasks assigned to a workstation have to be finished before the takt time deadline. Then the aircraft is moved from each workstation to the next one and so on until the aircraft assembly process is finished.

The advantages of Pulse Assembly Lines:

- Less investment is required.

- Increased flexibility and fault tolerance.
- In plant logistics is facilitated.
- More variety of work performed by operators.

The disadvantages of Pulse Assembly Lines:

- The planning and execution of tasks have to be accurately supervised, since a deadline has to be met at each workstation, the procedures for the management of "remaining work", the tasks that could not be finished at the assigned workstation have to be implemented.
- If tasks using expensive equipment are assigned to different workstations, it should be doubled or its usage should be synchronized.

The following picture shows an example of Pulse Assembly Line (Fig. 1.1).

Figure 1.1: Lockheed Martin F22 Pulse FAL. Source: Lockheed Martin.

Currently, it is the most used organization of assembly lines in aircraft industry. This is the type that is considered in this thesis.

1.2.3 Moving Assembly Line

This organization is closed to the previous one but the separation of workstation zones is less separated. Instead of total quite long takt time, the aircraft is moved almost constantly (with a short time interval) but for shorter distances. The equipment and workers are installed along the line in order to perform the assembly tasks.

The main advantage of Moving Assembly Line is the effective flow of material and working personal. The main disadvantage of Moving Assembly Line is its low tolerance to uncertainties. In case of a breakdown, the entire assembly line has to be stopped. This also means that all emergency situations should be solved fast.

Moving Lines are using in assembly process of Boeing 737, 747 and 777 (Fig. 1.2).

Figure 1.2: Boeing 777 Moving FAL. Source: Boeing.

1.3 Scheduling issues in aircraft assembly lines

Optimistic forecasts for future world air traffic resulting in constantly growing demand in this industrial sector put a lot of pressure on the efficiency and reliability of aircraft final assembly lines [12, 13, 14].

Due to the technical and financial reasons, aircraft assembly lines are characterized by a relatively small number of workstations, but a huge number of tasks to be performed at each workstation and a relatively long takt time. By these characteristics, they differ considerably from automotive assembly lines where the number of workstations is usually very high, but the number of tasks per workstation is limited because of a short takt time. For this reason, the optimization planning techniques developed for automotive assembly environment cannot be applied directly for aircraft assembly and have to be adapted to this particular manufacturing environment.

Moreover, during the aircraft assembly process operators perform tasks manually in postures related to high ergonomic risks. To improve the performance of aircraft assembly lines, both economically and socially, it is necessary to develop new scheduling methods taking into account both economic and ergonomic criteria operators. However, only recently, works of [15, 16, 17] presented the first scheduling methods explicitly integrating the ergonomic data. All cited studies were realized in the context of automotive production were the task repetivity is much higher than in aircraft assembly lines. A comprehensive survey on the integration of ergonomic criteria in planning methods for assembly lines was recently presented in [18]. It showed that the ergonomic conditions may be improved without or with a little impact on the economic performances of the assembly lines on the short term while the realized investment brings very positive results on the long term.

The objective of this thesis is to consider both economic and ergonomic aspects in scheduling of tasks in aircraft assembly lines taking into account the following requirements:

- optimal usage of resources, including time, space, special equipment, human resources;
- minimization of operator errors;
- takt time reduction;
- improvement of working conditions.

It has been mentioned that in contrast to the intensively studied automotive assembly lines, in aircraft assembly line context, an important number of operators and a huge number of tasks have to be assigned to a low number of workstations. The problem of task scheduling at each workstation can be considered as a special case of Resource Constrained Project Scheduling Problem (RCPSP). The difficulty is that even in its classic formulation RCPSP proven to be NP-hard in the strong sense [19].

Solving RCPSP is related to selecting task execution modes and correct assignment of processing intervals of tasks. In this problem, the availability of resources is necessarily assumed to be limited; therefore precedence relations and resource constraints have to be taken into account.

Two types of precedence relations are considered. In the classic RCPSP statement, precedence relations establish a predefined sequence between two activities, the second activity can start only when the first has been completed. Generalized precedence relations allows to define time lags between the start times of two activities. RCPSP statement with time lags refers in the literature to RCPSP/max.

Set of renewable resources represents inputs like equipment, human resources that are used, but not consumed when performing the project.

Moreover, for each particular context, additional specific constraints can be formulated. In this research, we pay a lot of attention to ergonomic and takt time constraints.

Indeed, aircraft assembly process includes a large variety of manual operations in postures related to high ergonomic risks. In ergonomics, numerous studies were realized on occupational hazards, but their results are not integrated in planning horizon. Usually companies conduct occupational risk assessments for existing units of work (in France according to Labour Code Articles L.4121-2 and L.4121-3). This assessment is designed to identify a broad set of risks. Our objective is to use existed scoring methods at planning stage to avoid high ergonomic risks. The objective of this study is to integrate the ergonomic models in appropriate planning methods in the context of aircraft assembly lines.

1.4 Manuscript organization

The thesis is organized in the following manner.

Chapter 2 presents a state of the art in the field of the evaluation of ergonomic risks as well as scoring methods already employed in scheduling applications for production systems.

Chapter 3 overviews the different formulations of RCPSP, its generalizations, the techniques for Lower Bound calculation and the existing solution methods.

Chapter 4 proposes a new pseudo-polynomial algorithm to find a makespan lower bound for RCPSP with time-dependent resource capacities. Its idea is based on a consecutive evaluation of pairs of resources and their cumulative workload. A generalization for continuous time and piece-wise constant capacity function is also developed.

Chapter 5 presents new constraint propagators for RCPSP based on resource capacity and time lags propagation in order to accelerate its solution with constraint programming approach.

Chapter 6 presents new formulations integrating physical and cognitive parameters of tasks in RCPSP. The solution method consists of two parts. The first part uses CP to solve the scheduling problem with the aggregated demand. The objective is to find the schedule with the minimal makespan subject to resource constraints and precedence relations. The second part assigns the operators to the scheduled tasks under the ergonomic constraints. The second part is solved with Integer Linear Programming (ILP) approach. The results obtained in numerical experiments are presented and analyzed.

Chapter 7 considers an upper level optimization problem where not only the tasks have to be assigned to the operators and scheduled but also the number of operators and their skills have to be defined. A new ILP formulation is developed and the first results are obtained for industrial case studies.

Finally, Chapter 8 presents general conclusions and proposes research perspectives to continue studies in the field.

Chapter 2

Evaluation of ergonomic risks in production systems: State of the art

One of the primary goals of production managers is to provide safe working conditions for their employees in order to preserve their health, to improve their well-being and to reduce absenteeism. Exposure to hazards in the workplace can cause or contribute to the risk of developing work related diseases, e.g. musculo-skeletal disorders (MSD). According to Occupational Safety and Health Administration (OSHA, USA, [20]), the following factors are mostly responsible of MSD: force, repetition, awkward postures, static postures, quick motions, compression or contact stress, vibration. Moreover, often these factors act in combination while substantially increasing the likelihood of causing an MSD [21]. In order to identify the occupational risk factors, OSHA advices to evaluate the following aspects of work: physical demands of work, workplace and workstation conditions and layout, characteristics of object(s) that are handled or used and environmental conditions. The following psychosocial factors should be also analysed: work pace, task demands, autonomy, monotony, social support, work/rest cycle, job uncertainty. A study of German pension fund found that the relative risk of early retirement increased by 67% as a result of both physical and psychosocial risk factors compared to the exposure of physical risks only [49]. It should be also noted that OSHA ranks assemblers as number 4 in the top ten occupations concerned by MSDs [20].

2.1 Physical ergonomic methods

To evaluate the ergonomic risks, a number of direct methods, observational methods, subjective methods, and other psychophysiological methods have been developed by ergonomists. The integration of such approaches in planning or scheduling procedures is not always easy because of the complexity of assessment.

Among the most commonly used methods, the following have been already related to scheduling problems in production systems in the literature:

- NIOSH-Eq – *National Institute for Occupational Safety and Health lifting equation* [22]. The National Institute for Occupational Safety and Health (NIOSH) developed the Work Practices Guide for Manual Lifting in 1981 in order to assist safety and health practitioners in the assessment of the risks related to lifting and lowering jobs with the objective to prevent and reduce the low back pain among workers and the related injuries. This Manual was revised and extended in 1991 and it is currently a part of ISO 11228 standard.

- LI – *Lifting Index* [23]. The lifting index displays the ratio between the handled load and a recommended load. The latter depends on the weight of the handled object, horizontal and vertical locations, distance, angle of symmetry, frequency of lift, duration and the coupling between hands and the object.

- RULA – *Rapid Upper Limb Assessment* [24]. The RULA tool offers worksheets for rapid assessment of ergonomic risks of upper limbs, neck and trunk. It consists of one worksheet and requires just seven steps to compute the final score. The final score depends on the applied forces, awkward and static postures and the frequency of repetition.

- REBA – *Rapid Entire Body Assessment* [25]. REBA is a method specially designed to be sensitive to the type of unpredictable working postures found in health care and other service industries. It was developed by a team of ergonomists physiotherapists, occupational therapists and nurses collected and individually coded over 600 postural examples.

- OCRA – Occupational Repetitive Action tool [26]. The OCRA index evaluates repetitive handling at high frequency performed by upper limbs and it is calculated separately for each hand. The final OCRA index is computed as the ratio between the actual and the recommended frequency of actions measured as the number of repetitions per minute. The recommended frequency of actions depends on the applied forces, postures and additional risk factors, such as vibration.

- JSI – *Job Strain Index* [27]. The JSI uses a methodology similar to that of OCRA with two additional parameters: speed of work and duration of strain.

- DND – *Daily Noise Dosage* [28]. DND provides recommended exposure limit for occupational noise depending on time of affection, sound volume and frequency.

- EAWS – *Ergonomic Assessment Work Sheet* [29]. The EAWS tool assesses postures, action forces, manual material handling as well as other whole-body risk factors and repetitive loads of the upper limbs. The results of the EAWS estimation are two aggregate risk values: risk points for the whole body and risk points for upper limbs. Higher risk points indicate higher risks for musculoskeletal disorders.

- EnerExp – *Energy Expenditure Method* [30]. Energy Expenditure Method based on the amount of energy spent by a worker during activity execution.

Table 2.1 shows that different ergonomic methods consider different risks. Existing researches which consider line balancing and scheduling problems subject to human factors in assembly systems were summarized in [18] and presented in Tab. 2.2 and 2.3 respectively.

Table 2.1: Ergonomic methods comparison.

method	focus	individuality	neck	trunk	hands	legs	whole body	noise
NIOSH-Eq	lifting task						+	
LI	lifting tasks	+					+	
RULA	postures		+	+	+			
REBA	postures		+	+	+	+		
OCRA	upper extar-mities				+			
JSI	upper extar-mities				+			
DND	noise							+
EAWS	general risk assesment	+	+	+	+	+	+	
EnerExp	general risk assesment	+					+	

Therefore to improve the performance of workers and decrease the impact on their health it is reasonable to use several methods in the same time. We follow this idea in our problem statement.

In practice, the evaluation of ergonomic hazards is often performed manually and it is a time consuming procedure. Recently, simulation tools [93] and computer vision tools [94] have been developed in order to evaluate some ergonomic parameters automatically with promising precision. In order to input ergonomic parameters in planning and scheduling procedures, the majority of existing approaches use an off-line evaluation of tasks.

Another way to assess the ergonomic risks of tasks is to evaluate the muscle fatigue generated by their execution. Fatigue is considered as an important factor in evaluating the cost of providing effort, as well as in decreasing the productivity. Additionally, numerous studies have recognized work-related fatigue as a substantial cause of cognitive

Table 2.2: Summary of the contributions to the assembly line balancing that consider physical ergonomic risks. Notes: Parameters – general job-specific physical demand parameters, B&B – branch-and-bound algorithm, GA – genetic algorithm, LS – local search, SA – simulated annealing, GRASP – greedy randomized adaptive search procedures, heuristic – other construction heuristic. Dash is used if no customized solution method was proposed, for example, if the article utilized an off-the-shelf solver.

Reference	Measurement of ergonomic risks	Ergonomic risks considered as... Constraints	Obj. f	Solution method
Gunther et al. [50]	EnerExp	x	x	B&B
Carnahan et al. [15]	Fatigue (Woods et al. [51])		x	GA, heuristic
Jaturanonda and Nanthavanij [52]	RULA		x	Heuristic (LS)
Choi [53]	Environmental parameters, awkward and static postures, force loads (check-list of 13 risk factors)		x	–
Otto and Scholl [16]	NIOSH-Eq, OCRA, JSI, EAWS	x	x	Heuristic (SA, LS)
Rajabalipour Cheshmehgaz et al. [54]	Awkward and static postures		x	GA
Mutlu and Ozgormus [55]	Parameters	x		–
Xu et al. [56]	Force loads, repetitiveness, vibration (ACGIH [57])	x		–
Jaturanonda et al. [58]	RULA		x	Heuristic (LS)
Kara et al. [59]	Task rigidity, energy expenditure, quality of illumination	x		–
Sternatz [60]	Awkward and static postures, force loads (internal method of a firm)	x		Heuristic
Barathwaj et al. [61]	RULA		x	GA
Akyol and Baykasoglu [62]	OCRA		x	Heuristic
Battini et al. [17]	EnerExp		x	–
Bautista, Alfaro-Pozo et al. [63]	OCRA, RULA, NIOSH-Eq		x	GRASP
Bautista, Batalla-Garcia et al. [64]	Parameters	x	x	–

Table 2.3: Summary of the contributions to the job rotation scheduling that consider physical ergonomic risks. otes: Parameters – general job-specific physical demand parameters, B&B – branch-and-bound algorithm, GA – genetic algorithm, LS – local search, SA – simulated annealing, GRASP – greedy randomized adaptive search procedures, heuristic – other construction heuristic. Dash is used if no customized solution method was proposed, for example, if the article utilized an off-the-shelf solver.

Reference	Measurement of ergonomic risks	Ergonomic risks considered as… Constraints	Obj. f	Solution method
Carnahan et al. [65]	JSI-L		x	GA
Nanthavanij and Kullpattaranirun [66]	DND		x	GA
Tharmmaphornphilas et al. [67]	DND		x	–
Yaoyuenyong and Nanthavanij [68]	DND		x	Heuristic(LS)
Tharmmaphornphilas and Norman [69]	JSI-L, DND		x	–
Kullpattaranirun and Nanthavanij [70]	DND		x	GA(LS)
Asawarungsaengkul and Nanthavanij [71]	DND	x		–
Yaoyuenyong and Nanthavanij [72]	DND	x		Heuristic(LS), B&B
Tharmmaphornphilas and Norman [73]	JSI-L		x	Heuristic(LS)
Seckiner and Kurt [74]	Parameters		x	SA
Asawarungsaengkul and Nanthavanij [75]	DND	x		GA(LS)
Asawarungsaengkul and Nanthavanij [76]	DND	x		–
Seckiner and Kurt [77]	Parameters		x	ACO
Yaoyuenyong and Nanthavanij [78]	Parameters, DND, EnerExp	x		Heuristic(LS)
Aryanezhad et al. [79]	JSI-L, DND	x	x	–
Diego-Mas et al. [80]	Force loads, awkward and static postures, repetitiveness, capacities of workers		x	GA
Michalos et al. [81] and Michalos et al. [82]	Fatigue, repetitiveness		x	Heuristic
Nanthavanij et al. [83]	DND	x		Heuristic (LS)
Asensio-Cuesta, Diego-Mas, Canos-Daros, et al. [84]	Force loads (Rodgers [85])		x	GA
Asensio-Cuesta, Diego-Mas, Canos-Daros, et al. [86]	OCRA, monotony		x	GA
Michalos et al. [87]	Fatigue (Ma et al. [88]), repetitiveness		x	Heuristic
Otto and Scholl [89]	EAWS		x	Heuristic(LS), tabu search
Mossa et al. [90]	OCRA	x	x	–
Yoon et al. [91]	REBA		x	–
Song et al. [92]	NIOSH-Eq, force loads (Rodgers [85]), geometry of tasks	x		GA(LS)

errors and increased safety risks [95]. A critical comparison of muscle fatigue models is presented in [88] and [96]. Recently, a general fatigue disutility model was proposed in [97]. In this research, we will consider physiological aspects of ergonomic risks in detail.

2.2 Physiology of muscular activity

A lot of assembly tasks are related to muscular activity. If we want to consider physical ergonomic aspects and operate with the terms "fatigue" and "muscular pain", we have to find the reasons of these phenomena. In this section, the processes involved in muscular activity on cellular-molecular level are described. First of all, let us consider the structure of muscle tissue.

2.2.1 Muscle fiber structure

There are three types of muscle tissue in human organism: skeletal, smooth and cardiac. In this research, only skeletal muscles are considered because they are connected with the performance of mechanical activities of workers. Muscle is a number of fascicles each of which consists of fibres (muscular cells). The detailed structure of the muscle tissue is shown in Fig. 2.1.

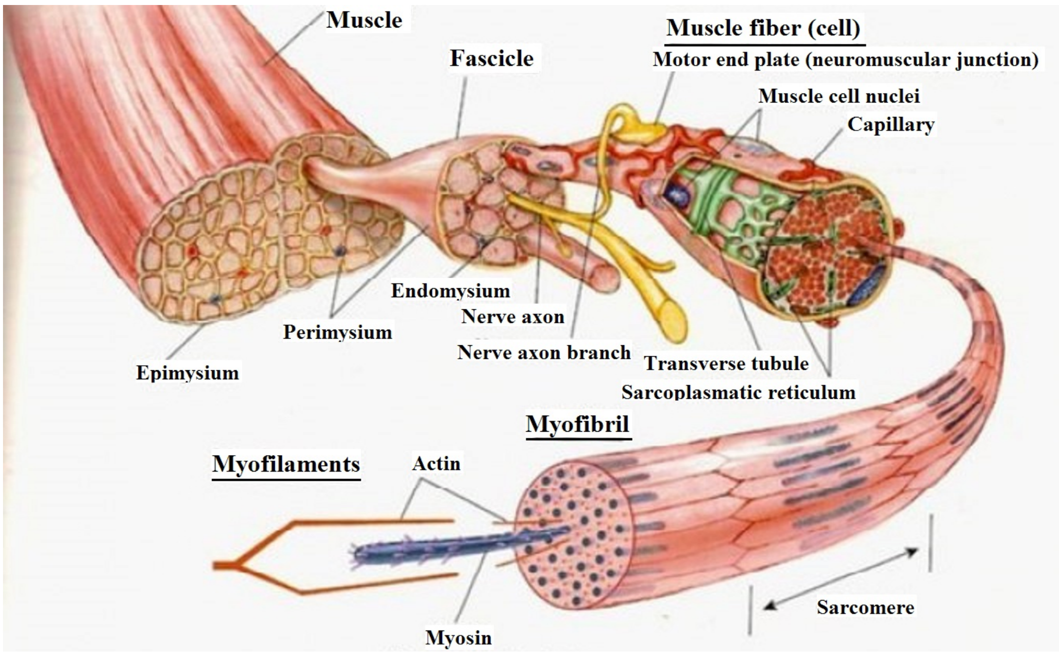

Figure 2.1: Muscle tissue structure.

2.2.2 Description of muscular activity process

Let us briefly describe the muscular activity process. Direct work of muscular fibres are carried out by "myofibrils" – inner structures of muscular cells. The whole process was discovered by Nobel Prize winner Andrew Fielding Huxley [98] and can be shortly described as follows.

Actin filaments begin sliding motion relative to the filaments of myosin to the center of the sarcomere 2.2. In this case, the filament of myosin is surrounded by 6 threads of actin, and the filament of actin is surrounded by 3 strands of myosin. The movement of myosin filaments is possible due to the fact that the myosin filament has lateral branches, so-called bridges.

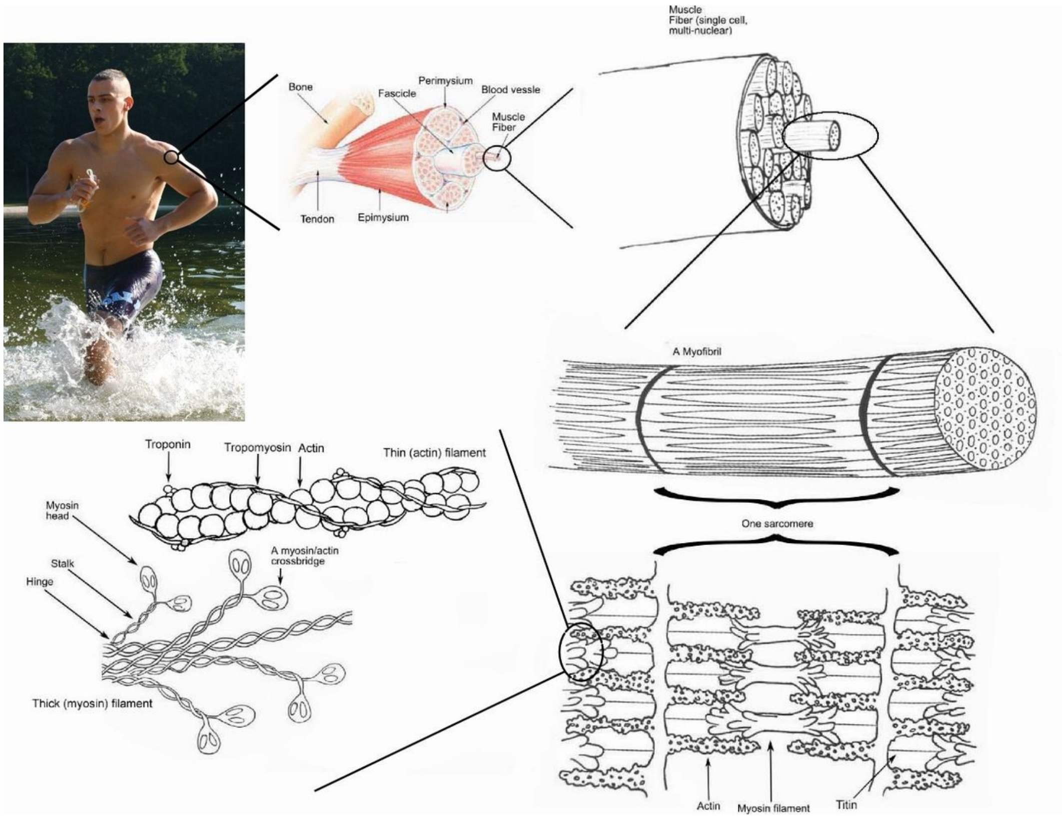

Figure 2.2: Actin filament and myosin strands.

Between the filaments of actin and myosin bridges can be created and destructed, but for destruction of bridges energy of adenosine triphosphate (ATP) is required. Therefore bridges can turn, i.e. myofibryl size decreases and muscle works. Note that the bridges are formed in the presence of calcium ions in the sarcoplasm and calcium ions (Ca^{++}) are released from the tank only by incoming electric impulse.

2.2.3 Energetic aspects of muscular activity

As it was mentioned before, ATP energy is required for actin-myosin bridges turns. To break a bridge, ATP loses one P-atom and becomes adenosine diphosphate (ADP). In the common case, to break the bridge again it should be restored to ATP. In the research [99], it was shown that ATP is too large to move freely around the cell. The resynthesis of myofibrilar ATP occurs at the expense of creatine. Creatine has the ability to both attach a phosphate residue to itself, turning into creatine phosphate (CrP), and give it away, forming a free creatine (Cr). In both forms, creatine freely moves through the cell and passes through the membrane of myofibril. Giving its phosphate group ADP, CrP restores it again to ATP:

$$ADP + CrP = ATP + Cr,$$

this well-known reaction was discovered by Karl Lohmann in 1934. After that free creatine Cr should be restored to CrP, using one of two ways:

- take P-group from the sarcoplasmic ATP;
- take P-group from mitochondrial ATP.

When Cr restored to CrP it can again be involved into Lohmann reaction to recharge myofibril ADP.

The choice of way to restore Cr to CrP depends on the type of muscle fibre, all of which are listed below.

- **Glycolytic muscle fibers (GMF).** There is almost no mitochondria, Cr can be restored only by sarcoplasmic ATP. Sarcoplasmatic ADP can be restored to ATP using anaerobic glycolysis (Fig. 2.3). Glycolysis of one molecula of glucose makes two moleculas of pyruvic acid and two ADP can be restored to ATP. Reduction of pyruvic acid by lactate dehydrogenase produces lactate, which is unstable acid and can be easily ionize a proton from the carboxyl group, producing the lactate ion.
- **Oxidative muscle fibers (OMF).** There is almost no sarcoplasmic ATP, and all myofibrils are surrounded by mitochondria (Fig. 2.4). Cr restores by mitochondrial ATP. Mitochondrial ADP can be restored to ATP by the "citric acid cycle" of reactions (Fig. 2.5), which use acetyl coenzyme A (acetyl-CoA), water and oxygen to produce carbon dioxide, water and restore mitochondrial ADPs. Acetyl-CoA can be obtained from pyruvic acid and as a result of beta-oxidation of fatty acids (Fig. 2.6).
- **Intermediate muscle fibers (IMF).** Mixed type of muscle fiber. There are some mytochondria, but not so many as in OMF. Both described ways can be used.

The most important point is that during the activity, anaerobic glycolysis produces lactate faster than it utilizes in citric acid cycle. Hence, amount of lactate in IMF increases during the activity.

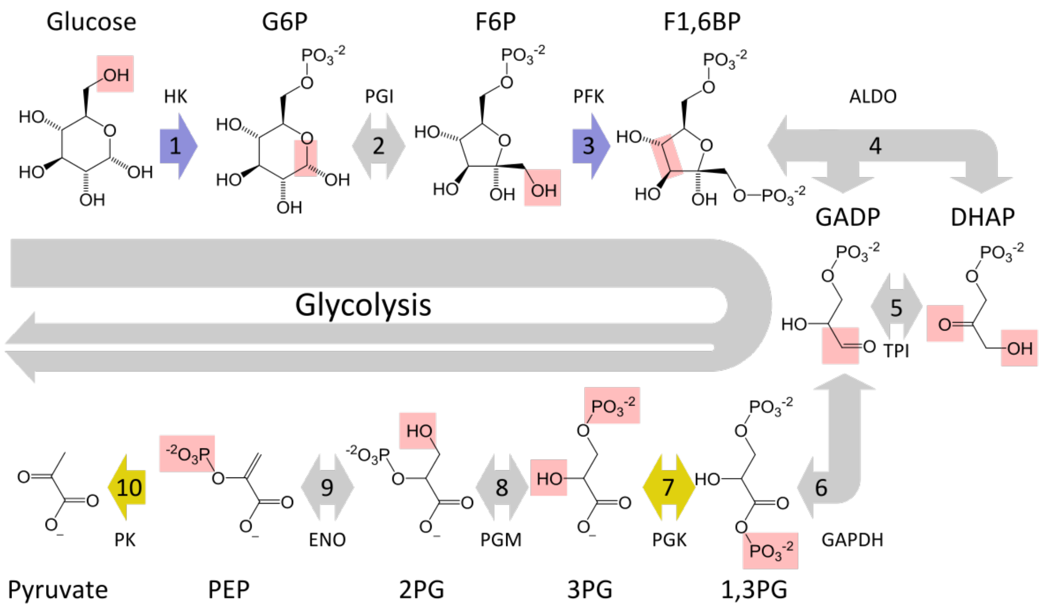

Figure 2.3: Anaerobic glycolysis.

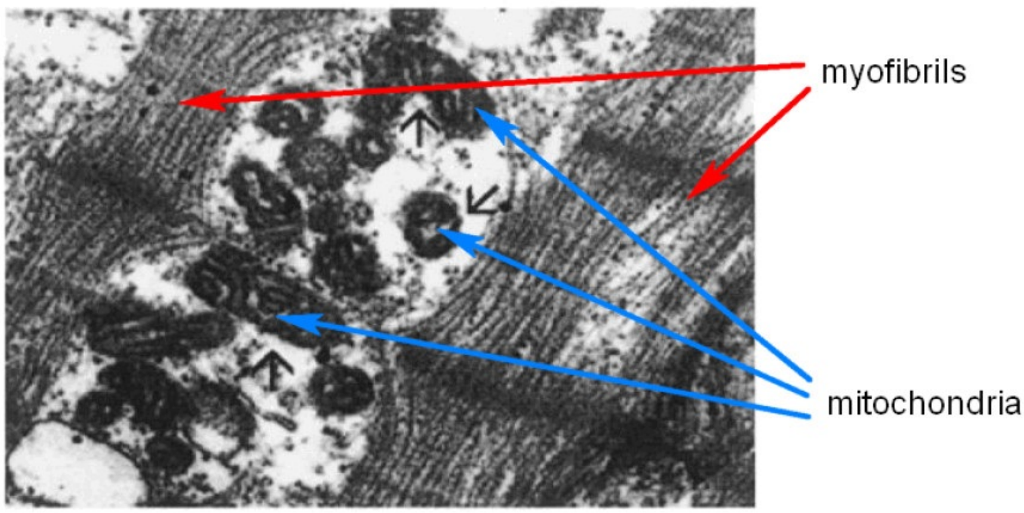

Figure 2.4: Mitochondria and myofibrils.

All considered processes take place in independent muscle fibres, but in each skeletal muscle all three types of MF are presented. Activation of each type of MF depends on the demanded power of muscle activity. Furthermore, glucose, oxygen and water should be

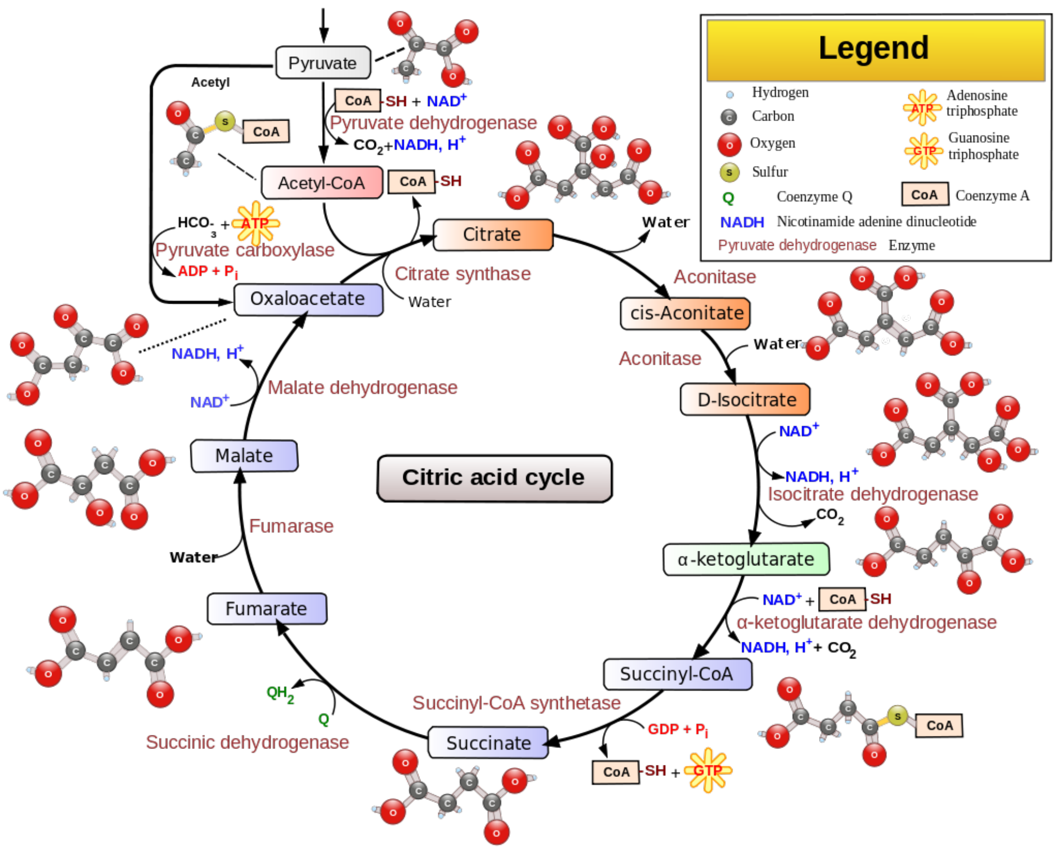

Figure 2.5: Citric acid cycle.

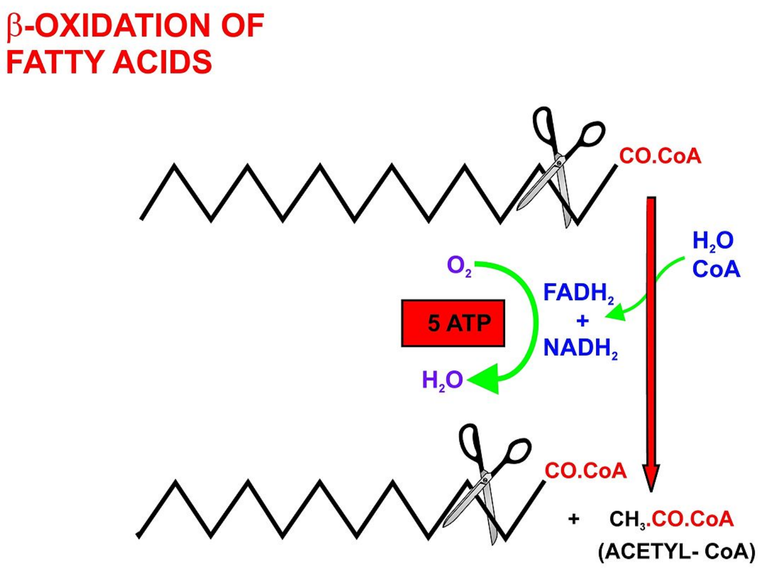

Figure 2.6: Beta-oxidation of fatty acids.

transported to MF by microcirculatory system, which functionality also can be affected by muscle activity. Therefore the real situation is more complex.

2.2.4 Initiation of muscle working process

As it was mentioned before to initiate muscular activity, electric impulse should release calcium ions (Ca^{++}) from the tank. This process is managed by motoneurons – nerve cells, whose bodies lie in the spinal cord, and long branches – axons in the motor nerve approach the muscle. Entering the muscle, the axon branches into many branches, each of which is brought to a separate fiber. Thus, one motoneuron innervates a whole group of fibers, which works together. The system, which includes motoneuron, axon and a set of muscle fibers, was called the "motor unit" (MU) (Fig. 2.7). The muscle consists of a set of MU and is able to work not with its entire mass, but in parts, which allows us to regulate the force and speed of contraction.

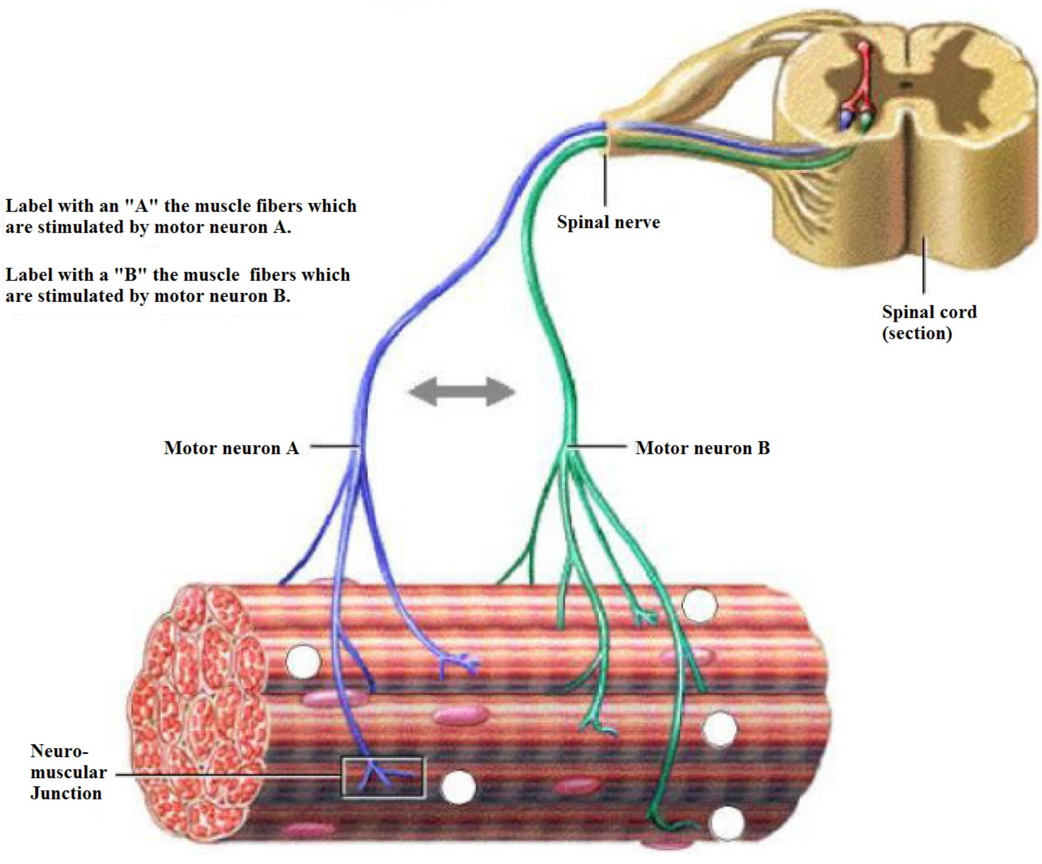

Figure 2.7: Two different motor units.

Motoneuron can be activated by a signal from cerebral cortex. Each MU has its own threshold of excitation, which directly depends on the size of the motoneuron. If the excitation is below the threshold, the MU is not active. If it is at the threshold level or above, the MU is excited, and all its MFs work with the maximum possible power for them by the "all or nothing" principle. If the MU threshold is 20 Hz and when the pulsation is below 20 Hz, the MU is not active, when it reaches 20 Hz it activates and shows the maximum power and further increasing the ripple frequency to 50 or 100 Hz does not introduce any changes into its operation.

Threshold levels of motoneurons connected to different types of muscle fibres can be sorted in ascending order as follows: $OMF < IMF < GMF$. This means that OMF activates even for easy muscle activity and GMF activates only when very large force is required.

2.2.5 Muscular activity modeling

Let us consider a muscle that makes a simple dynamic task. Since such a task requires small force, only MUs with low threshold level will be activated. This MUs activate some OMFs, which start working.

1. 0s – 0.02s. Actin filaments begin sliding motion by creating and destroying actin-myosin bridges. The first bridges can be destroyed using an energy of myofibril ATP, and first two turns can be done by all actin filaments and activated MF works with maximum intensity (100%).

2. 0.2s – 15(20)s. The ATP's energy restored by CrP can be used. It becomes Cr and starts working as a shuttle. Not all ATP can be restored in time, not all bridges can be destroyed, hence the intensity decreases to 90%.

3. 15(20)s – 40(60)s. All Cr work in shuttle mode, this decreases the intensity to 50%. Mytochondrial ATP restores by the "citric acid cycle".

4. 40(60)s – .... Citrate (one element of "citric acid cycle") inhibits glycolysis, beta-oxidation of fatty acids becomes the main mechanism for the energy supply. Recharging of mytochondrial ADP becomes slower, and the intensity of MF decreases to 40-45 %.

The graphic of intensity is presented in Fig. 2.8. During the process, the intensity of the considered MF activity decreases and the central nervous system (CNS) increases the frequency of impulses to ensure the required power. This leads to the involvement of new MU. This stepwise process involves more and more MU until the moment when involved MF can ensure the required power even working with 40% of intensity. If only OMF are involved in this process, muscular activity can be carried on for a long time until there are enough intramuscular fats to supply the process.

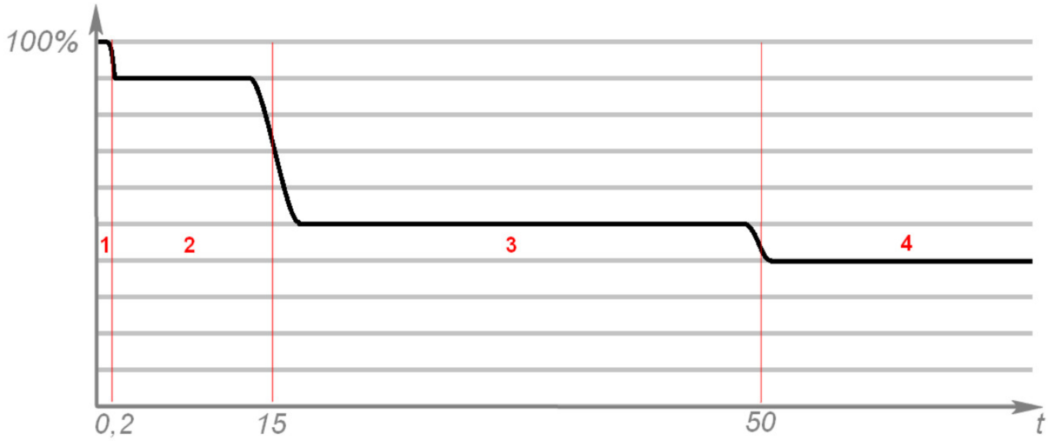

Figure 2.8: OMF intensity.

After all fats have been used, the process switches to the glycolysis supply, and can be done until there is enough glucose. We can consider a quiet walk as an example of such a process. An indication of the recruitment of all OMF is an increased level of lactate in blood and an increased pulmonary ventilation. Pulmonary ventilation is enhanced, due to the formation and accumulation of hydrogen ions in the IMF, which, when released into the blood, interact with the buffer systems of the blood and cause the formation of excess (non-metabolic) carbon dioxide. If OMF power is not sufficient to ensure the required power, CNS increases the frequency of impulses to involve MU related to IMF. Intermediate muscle fibers are those in which the mitochondrial masses are insufficient to balance the formation of pyruvic acid and its oxidation in the mitochondria. In IMF, mitochondrial mass is insufficient to balance the formation of pyruvic acid and its oxidation in the mitochondria. In IMF, after a decrease in the concentration of CrP, glycolysis is activated, part of pyruvic acid begins to convert to lactic acid, which leaves the blood, penetrates into the OMF. Lactate is the strongest inhibitor of fat oxidation. Lactate reduces the use of fat acids by enhancing non-esterification, with no effect on lipolysis. The entry of lactate into OMF leads to inhibition of fat oxidation, carbohydrates become the main substrate. First of all, OMF uses lactate, as the most economically profitable source of energy. Lactate also can be used as a substrate by heart and respiratory muscles. If the intensity of this process allows to hold the concentration of lactate in the blood lower than 4 mM / l, this process can be carried on for a long time until there is enough glucose supply. If the produced power is still not enough, new MFs are involved.

If the activity needs large power, GMF are involved in the process. This enhances the processes of anaerobic glycolysis, more lactate and hydrogen ions are released into the blood (and its pH level is decreased). When lactate enters the OMF, it turns back into pyruvic acid, but the thickness of the mitochondrial OMF system has a limit. Therefore, firstly there comes a limiting dynamic equilibrium between the formation of lactate and its consumption in OMF. If involved IMF and GMF produce more lactate than mitochondrial system can use, equilibrium is disrupted. This is accompanied by a further increase in pulmonary ventilation, heart rate and oxygen consumption.

But one of the most important point is that not all lactate molecules can be removed from GMF by blood. Lactic acid is an unstable compound and easily dissociates into lactate anion La^- and hydrogen cation H^+. Lactate is a large molecule, it cannot participate in chemical reactions without the participation of enzymes, so it cannot damage the cell. But H^+ is the smallest positively charged atom, therefore it can penetrate complex cellular structures and leads to significant chemical damage. H^+ cations is the most important cause of muscle fatigue. In myofibrils, H^+ is added to the troponin which interferes Ca^{++} ions to join it. As a result, the tropomyosin thread cannot move and the myosin bridges with the actin filaments cannot be constructed. The more H^+ penetrates into the myofibrils, the less number of myosin bridges are able to be involved in work. This leads to a decrease of MF contraction force. Cellular membranes do not release into the blood stream individual protons and anions, and only neutral molecules are released, therefore, hydrogen ions cannot enter the blood, but can only lactic acid. As noted earlier, almost immediately after the beginning of the involving GMF, lactic acids start to get into the blood from them. This lengthens the period of the onset of failure, but not for long. After 60 sec of the GMF work, so many hydrogen ions are accumulated, that they lose the force of contraction practically to zero. After the fall in the power of the muscle fibres involved, new ones are gradually incorporated. If there are not enough MF to carry out an activity, a failure occurs.

Another important impact of H^+ ions is the catabolic effect, related to the destruction of "lysosomes". Lysosome – is a membrane-bound organelle, spherical vesicles which contain hydrolytic enzymes that can break down virtually all kinds of biomolecules (Fig. 2.9). H^+ ions are able to destroy lysosome membrane and release hydrolytic enzymes in the sarcoplasm. High concentrations of hydrogen ions in the cell for a long time can lead to paranecrosis of the cell. This process is very painful. Such a process occurs in the heart of sportsman, which run with the high heart rate for a long time. Paranecrosis of myocardiocytes decreases its electrical conductivity for the whole remaining life, which leads to the increasing of the probability of myocardial infarction.

In comparison to the dynamic mode of muscular activity, static work has one important difference. During the muscular work in static mode active myofibrils compress the vessels of a capillary system, which leads to hypoxia and rapid accumulation of lactate. Without oxygen OMF and IMF can use only anaerobic glycolysis to restore ADPs. Partially hydrogen ions can be consumed by mitochondria but if H^+ concentration becomes too high, then mitochondria grows in size and bursts. If required power is not very high,

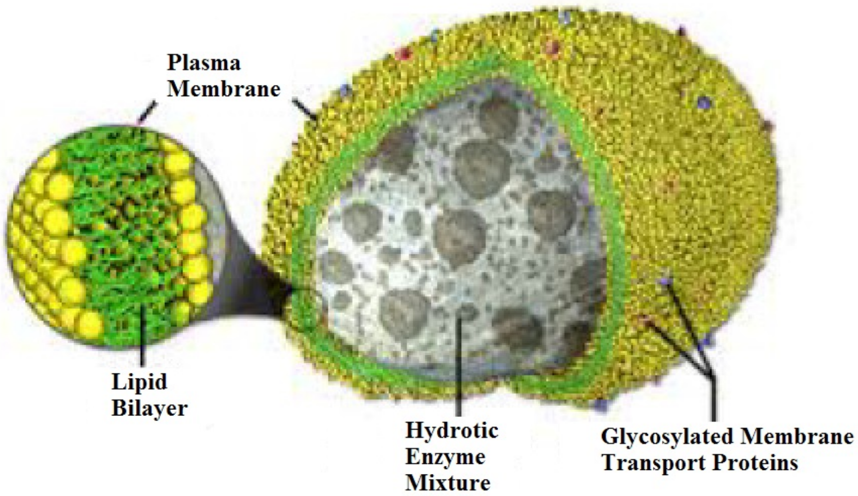

Figure 2.9: Lysososme structure.

failure does not occurs, because there are still enough non-active MF to be involved in the process to replace the disrupted ones. Only a burning sensation in the muscle is felt. Human can continue to do the work, but the process of catabolism will continue.

If the muscle is relaxed, in OMF and IMF H^+ ions turn into water in mitochondria, but in GMF high concentration of H^+ can persist for more than one hour. The best two ways to decrease the concentrations of H^+ ions are light dynamic activity in the mode which involves only OMF of the same muscle or a lot of OMFs of large muscle groups (i.e. walking).

Described processes lead us to the statement that one of the main objective to decrease muscular pains and fatigue is to avoid the long periods of high H^+ concentration in MF. Therefore, worker's schedule should alternate the activities which can increase and decrease H^+ concentration in each MF. In Tab. 2.4, different types of activities and their impact on H^+ concentration are presented.

Suppose that there is a set of activities A which should be carried on by the set of workers W. We can associate each activity $i \in A$ with the vector of functions $H^i(t) = \{h_1^i(t), \ldots, h_k^i(t)\}$, which represents the increasing or decreasing the hydrogen ions concentration in k considered MFs of different muscle groups during the processing of activity i by time t. Let worker $j \in W$ process activities under schedule π_j characterized by the consequence of pairs (activity, processing time): $\pi_j = \{(a_1, p_1), \ldots, (a_m, p_m)\}$. Start time s_i of job i equals to $\sum_{x=1}^{i-1} p_x$. Then, hydrogen ions concentration in the body of worker j at time t can be estimated as follows:

$$H_j(t) = H_j(0) + \sum_{i: s_i \leq t < s_{i+1}} H^i(p_i) + H_{i+1}(t - s_{i+1}).$$

Table 2.4: Impact of different types of activities.

type	activity characteristic	impact
1	Low power static activity	intermediate increasing of H^+ concentration in OMF and IMF
2	High power static activity	increasing of H^+ concentration in OMF, IMF; fast increasing of H^+ concentration in GMF
3	Low power dynamic activity	fast decreasing of H^+ concentration in OMF, IMF and GMF
4	High power dynamic activity	intermediate increasing of H^+ concentration in IMF, fast increasing of H^+ concentration in GMF
5	Low power dynamic activity of big muscle groups (active rest)	fast decreasing of H^+ concentration in OMF, IMF, intermediate decreasing of H^+ concentration in GMF
6	Passive rest	intermediate decreasing of H^+ concentration in OMF, IMF; slow decreasing of H^+ concentration in GMF

Then the negative impact of hydrogen ions concentration under the schedule can be evaluated by the following function:

$$F^i(\pi_j, t) = \int_0^t f(H_j(t))dt,$$

where $f(h)$ – negative impact by experiencing vector of hydrogen ions concentration h. The simplest way to evaluate negative impact is to consider the linear approximation, i.e. $h_l^i(t) = \alpha_l^i t$.

Therefore, the consideration of physical ergonomic aspects related to muscular activity in scheduling models has to take into account muscular pain and catabolism. Ergonomic constraints or objective criteria can be related to minimization of the negative impact, i.e. to decrease the time where worker's hydrogen ions concentration is higher then defined critical level CR, or cumulative catabolic effect (Fig. 2.10).

2.3 Conclusion

There is a wide range of existing ergonomic evaluation methods. These methods estimate the negative impact of working process on operator's health. Each method focuses on preventing a set of diseases and injuries. Some methods take into account operator's individual parameters, such as age, sex, height, weight. Therefore a set of different methods should be considered during the planning of FAL processing schedule to protect efficiently the operator's health. Physiological aspects prove that not only the set, but the sequence of processed tasks matters a lot in causing fatigue, muscle catabolism and pains. This

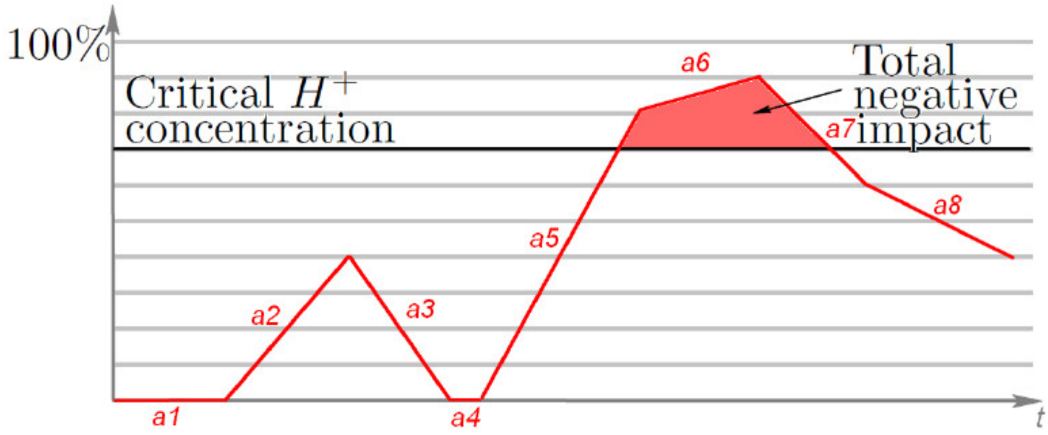

Figure 2.10: Cumulative catabolic effect.

leads us to idea that ergonomic impact should be calculated for short-time intervals, not only for the aggregated periods of working shifts.

Chapter 3

Resource Constrained Project Scheduling Problem: state of the art

3.1 Introduction

The Resource Constrained Project Scheduling Problem (RCPSP) is a well-known problem in scheduling theory. This problem is known to be NP-hard in the strong sense [19].

There is a plenty of different problems formulations and approaches to find optimal and suboptimal solutions for RCPSP. A comprehensive survey on project scheduling problems formulations and solution methods was presented in [31]. Benchmark PSPLIB was created [39], it is used to compare the performance of solution algorithms and lower bounds. Several comparisons of existing algorithms were conducted in the literature [32], [33], [34], [35], [36]. In this section, we recall the main achievements in the literature.

3.2 Basic RCPSP formulation

The basic formulation of RCPSP can be given as follows. There is a set of tasks N to be processed in the planning horizon T with the help of set of renewable resources R. The capacity of resource $r \in R$ equals to constant value c_r. For any task $j \in N$, the following parameters are given:

- p_j – processing time;
- a_{jX} – required amount of resource $X \in R$;
- r_j – release time, the earliest time from which task j can be started;
- D_j – deadline, the latest time for finishing task j.

Precedence relations between tasks are given by directed acyclic graph $G(N, E)$. If an edge $(i \rightarrow j) \in E$ exists, it means that task i must be finished before the starting time of task j.

Feasible schedule π is defined (constructed) for the sets of resources R and tasks N, if for any $j \in N$ *starting time* $S_j(\pi) \geq r_j$ is defined and all precedence and resource capacity constraints are satisfied. The set of all feasible schedules is noted by $\Pi(N, R)$. The objective is to find a feasible schedule with the minimal makespan (shortest project duration) i.e.

$$\min_{\pi \in \Pi(N,R)} \max_{j \in N} C_j(\pi), \tag{3.1}$$

where $C_j(\pi) = S_j(\pi) + p_j \leq D_j$ – the *completion time* of task j, where D_j – deadline of task j.

3.3 Additional notations

The basic formulation of the problem doe not take into account all aspects of real projects in different contexts. assembly process. To model additional features, the following notations should be introduced.

- The capacity of any resource $r \in R$ is defined by some non-negative piecewise constant capacity function $c_r(t)$.
- For any $j \in N$, we denote *the earliest starting time*, *the earliest completion time*, *the latest starting time* and *the latest completion time* by est_j, ect_j, lst_j and lct_j respectively.
- For any task j, the latest completion time can be defined by tail h_j, which should separate task completion time and project completion time;
- If $[lst_j, ect_j) \neq \emptyset$, then interval $[lst_j, ect_j)$ is a *compulsory part* of task j (firstly defined in [100]).
- The set of tasks $N_r \subseteq N$ which require resource $r \in R$ can be calculated, i.e. $N_r = \{j \in N | a_{jr} > 0\}$.
- The set of tasks $N^{CP} \subseteq N$ which compulsory parts are not empty can be calculated.
- The highest possible amount of resource $r \in R$ which can be used by non-compulsory parts of tasks is defined by piecewise constant function $c_r^n(t)$.

- The first and the last tasks of the project are dummy with zero processing times and without any resource requirements, i.e. $p_0 = p_{n+1} = 0$, $r_0 = r_{n+1} = 0$, $h_0 = h_{n+1} = 0$, $a_{0k} = a_{n+1k} = 0$ for any $k \in R$. Precedence relations with time lags are given by weighted directed acyclic graph $G = (N, E)$ where E – is the set of edges, defined bu triplets $\{i, j, e_{ij}\}$, where e_{ij} is the time lag between processing of tasks $i \in N$ and $j \in N$.

- If for a pair of tasks $(i, j) \in N^2$ disjunctive relation (firstly defined [101]) $g_{ij} = 1$ is defined, then processing intervals of these tasks do not intersect, i.e.

$$[S_i(\pi), S_i(\pi) + p_i) \cap [S_j(\pi), S_j(\pi) + p_j) = \emptyset.$$

- We use a disjunctive function $g_{ij}(t)$ defined on interval $[0, T)$. If tasks i and j can be processed simultaneously at time t, $g_{ij}(t) = 0$, otherwise $g_{ij}(t) = 1$. This formulation generalizes the concept of the disjunctive graph where $g_{ij}(t) = 1$ for each pair $i, j \in N^2$.

- Processing of tasks can require a set of operators O each of which has some specialities which belongs to the set S.

- Ergonomic impact can be measured by a set of methods M. For each triplet $(m \in M, j \in N, s \in S)$, the ergonomic impact of task j on operator with speciality s evaluated by method m is defined by erg_{mjs}. U_{mo} (U^h_{mo}) – is the critical level of the total ergonomic impact evaluated by method m for all tasks processed by the same operator o, which should not be violated in the planning horizon. Note that this critical level can depend on individual capacities of the operator. $U^i_{mo}(t_1, t_2)$ – an upper bound on total ergonomic impact evaluated by method m for all tasks processed by operator o in time interval $[t_1, t_2)$, where $t_1, t_2 \in [0, H)$ and $t_1 < t_2$.

- Set of tasks N can be divided into set of groups, i.e. for any task $j \in N$ can be defined group $g_j \in G$.

This generalizations will be used in constructed models depending on considered problem constraints and objectives. Presented notations will be reintroduced in problem statements to make it easier to understand. All of them you can find in the notations section.

3.4 Calculation of lower bounds on makespan

In the literature, a number of algorithms to calculate lower bounds on the makespan were proposed. There is a large number of publications devoted to the discussion on lower bounds for RCPSP. Recent comprehensive reviews can be found in [40] and [41].

The analysis of the computational complexity of some algorithms and the quality of the obtained bounds are discussed in [42]. The majority of efficient algorithms can be referred to as "destructive" methods. Such an algorithm starts with a defined project deadline T and tries to find a feasible schedule for it. If a feasible solution does not exist, the deadline is increased (usually by incrementing the deadline with 1 unit of time) and the calculation procedure restarts. The calculation continues until the algorithm cannot reveal any contradiction with the defined deadline or until the end of the allocated calculation time. Similarly, constraint programming methods can be applied as for example in [102] and [44].

Here below are shortly discussed the most effective methods for finding a lower bound on the makespan for RCPSP.

1. *Disjunctive bounds*

In the study of [103], each renewable resource is considered as a system of several identical processors with the number of processors equal to the capacity of the resource. The problem is formulated using mixed integer programming and heuristics are used to solve it.

Several algorithms [104, 105, 106, 107, 108] aim to facilitate the lower bound calculation by constructing complementary precedence relations with the verification of the assumptions that certain requirements of set A must / cannot be satisfied before / after some considered set of requirements B. These algorithms have polynomial runtime for one iteration, but the number of iterations increases exponentially when the number of requirements A and B increases.

2. *Cumulative bounds*

The algorithm of [109] is based on the identification of such sets of tasks for which the available amount of resource is insufficient for their parallel execution. Each resource is considered as a system of several identical processors where the number of processors per resource is less than its capacity. In the further studies [110] and [111], it is assumed that each processor can execute more than one task per unit of time and the same task can be completed by several processors.

Polynomial time algorithms were proposed to solve a relaxed problem [112, 113, 114] where the planning horizon (between the starting point and the deadline) was split into intervals and each resource was considered as a multiprocessor system with the number of processors equals to the capacity of the resource. Further, the interruption of tasks at the boundaries of intervals was allowed.

The relaxation to so-called Cumulative Scheduling Problem was also explored. It is obtained from the initial problem by ignoring all resources except one and replacing the precedence relations by release times and deadlines. The optimal makespan for this problem provides a lower bound for the initial problem. However, the obtained Cumulative

Scheduling Problem is also NP-hard in the strong sense. Nevertheless, methods developed for the calculation of a lower bound on the makespan for such a formulation provide a lower bound for the makespan of the initial problem as well [115, 116, 117, 118]. Satisfiability tests (SAT) can also be performed by dividing the planning horizon into intervals and checking the amount of the available resource in each of the considered intervals [107, 119, 120, 121].

3. *Methods based on Constraint Programming.*

Among the studies using different techniques of Constraint Programming, for example [102] and [44], the techniques developed in [122] and [123] are based on reducing the time intervals calculated for each task due to the analysis of the available amount of each resource for a chosen set of tasks. Such algorithms are time consuming because of the large number of possible sets under consideration.

4. *Algorithms based on the exploration of multi-resource constraints.*

Such algorithms [124, 125, 126] are based on the research of "critical sets" (MCS - minimum critical set, FS - forbidden set) i.e. sets of tasks that cannot be performed simultaneously because of resource capacity constraints, while any subset of such a critical set does not violate resource constraints and can be performed simultaneously.

5. *Linear programming relaxations.* A lower bound can also be obtained by a relaxation of the initial problem to a linear programming problem [127, 128].

3.5 Constraint programming to solve RCPSP

There is a wide range of existing constraint programming algorithms which can be applied to find an optimal/suboptimal solution for RCPSP or to make the problem easier to solve. Constraint programming is widely used to solve scheduling problems, including RCPSP. The most comprehensive surveys can be found [43], [44] and [45]. This research focuses on constraints based on resource usage, precedence and disjunctive relations.

The term of *disjunctive graph* was proposed in [101]. Algorithms to solve scheduling problems using disjunctive graphs were presented in [104].

A lot of constraints can be obtained by considering Cumulative Scheduling Problem (CuSP) – a single-resource version of RCPSP. The term *compulsory part* of a task was proposed in [100]. In [129], [130] and [123], compulsory parts were used to calculate *time-tables, resource profiles* and *resource histograms.* In papers [131], [132] and [133] time tabling sweep algorithms were presented to adjust task domains. In [134], the time tabling algorithm was improved by solving the rectangle placement problem. The best theoretical complexity of time table sweep algorithms were presented by Gay et. al. [135].

Resource-based Edge-Finding algorithms [122], [136], [137] can be used to make task domains tighter and to avoid resource overloads on time intervals. This approach can be improved by using Extended Edge-Finding algorithms presented by [138].

One of the advantages of constraint programming approach is the possibility to combine different algorithms for a more efficient propagation. In [139], [140] and [141], the synergy of Edge-Finding, Extended Edge-Finding and Time Tabling algorithms were used to obtain very good results.

Other propagators were discussed in [142], [43], [45] and in the surveys dedicated to the calculation of lower bounds on makespan [40] and [41].

3.6 Mixed Integer Linear Programming for RCPSP

One of the most popular approach for solving RCPSP is Mixed Integer Linear Programming (MILP). There is a plenty of MILP models for RCPSP, most of which are referred in surveys [37], [38]. MILP formulations are generally classified in three categories in accordance to modelling of time and resource-sharing. Let we briefly remind the most popular modeling methods.

3.6.1 Time-indexed formulations

This type of formulations (also refered in the literature as discrete-time) is Integer Linear Programming (ILP) formulations in which variables x_{it} indicating the status (i.e. started, in process, completed) of task i at time slot t. The number of variables is pseudo-polynomial, it linearly depends on the number of tasks n and the length of defined horizon T. Therefore, data pre-processing algorithms can be used for estimation an upper bound on horizon length, and to task domains tightening. These approaches can significantly decrease the number of unknown solution variables. There are three groups of variables using in Time-indexed formulations:

Pulse variables.

Pulse binary variables are commonly denoted by $x_{it} = \{0, 1\}$. If task i starts at time t, then $x_{it} = 1$, otherwise $x_{it} = 0$. The first model with pulse variables was firstly proposed in [127].

Let there is a set of tasks N to be processed in the time horizon T using the set of resources R. For each task $i \in N$ processing time p_i and required amount of resource $r \in R$ a_{ir} are defined. The capacity of resource $r \in R$ is defined by c_r. Precedence constraints are represented by the direct acyclic graph $G = (N, E)$, where E – set of edges. If edge (i, j) is defined, then task j can start its processing only when task i is completed. The last dummy task is indicated by the index $n+1$. The classic time-indexed formulation with pulse variables can be presented as follows.

- Objective function – makespan minimization:

$$\min \sum_{t \in T} t x_{(n+1)t},$$

subject to the following limitations.

- The variables are binary:

$$\forall i \in N, t \in T: \; x_{it} = \{0, 1\}.$$

- Precedence constraints:

$$\forall (i,j) \in E: \; \sum_{t \in T} t x_{jt} - \sum_{t \in T} t x_{it} \geq p_i.$$

- Resource constraints:

$$\forall r \in R, t \in T: \; \sum_{i \in N} \sum_{\tau = t - p_i - 1}^{t} a_{ir} x_{i\tau} \leq c_r.$$

- Each task should starts once:

$$\forall i \in N: \; \sum_{t \in T} = 1.$$

- If task domain $[r_i, D_i)$ is defined by release time r_i and deadline D_i for each $i \in N$, then the domain constraint can be formulated as follows:

$$\forall i \in N, t \in T \setminus \{r_i, \ldots, D_i - p_i\}: \; x_{it} = 0.$$

This formulation can be strengthened by using so-called *disagregated* precedence constraints [128]

$$\forall (i,j) \in E, t \in T: \; \sum_{\tau=0}^{t-p_i} x_{i\tau} - \sum_{\tau=0}^{t} \geq 0.$$

Step variables.

Binary variables denoted by $\xi_{it} = \{0, 1\}$. If task i starts at time t or earlier, the $\xi_{it} = 1$, otherwise $\xi_{it} = 0$. This type of variables firstly appeared in literature in [143]. Step variables are connected to pulse variables by formula $\xi_{it} = \sum_{\tau=0}^{t} x_{it}$ or $x_{it} = \xi_{it} - \xi_{it-1}$. In works [144] another type of step variables ξ'_{it}, which represents that the task i ends not later than at time t was considered for RCPSP modeling. The disagregated model with step variables can be formulated as follows.

- Objective function – makespan minimization:

$$\min \sum_{t \in T} t(\xi_{(n+1)t} - \xi_{(n+1)(t-1)}),$$

  subject to the following limitations.

- The variables are binary:

$$\forall i \in N, t \in T: \; \xi_{it} = \{0, 1\}.$$

- Precedence constraints:

$$\forall (i, j) \in E, t \in T: \; \xi_{i(t-p_i)} - \xi_{jt} \geq 0.$$

- Resource constraints:

$$\forall r \in R, t \in T: \; \sum_{i \in N} a_{ir}(\xi_{it} - \xi_i(t - p_i)) \leq c_r.$$

- Each task should starts once:

$$\forall i \in N, t \in T: \; \xi_{it} - \xi_{i(t-1)}.$$

- If task domain $[r_i, D_i)$ is defined by release time r_i and deadline D_i for each $i \in N$, then the domain constraint can be formulated as follows:

$$\forall i \in N, t < r_i: \; \xi_{it} = 0,$$

$$\forall i \in N, t \geq D_i - p_i: \; \xi_{it} = 1.$$

On-Off variables.

Binary variables $\mu_{it} = \{0, 1\}$. If task i is in process at time t, then $\mu_{it} = 1$, otherwise $\mu_{it} = 0$. Note that if task i ends at time t', then $\mu_{it'} = 0$. This type of variables was proposed in [145] and [146] for preemptive problems and in [147] for the RCPSP. On-off variables and pulse variables are connected by the following formulae:

$$\mu_{it} = \sum_{\tau = t - p_i + 1} x_{i\tau}, \tag{3.2}$$

$$x_{it} = \sum_{k=0}^{\lfloor t/p_i \rfloor} \mu_{i(t-kp_i)} - \sum_{k=0}^{\lfloor (t-1)/p_i \rfloor} \mu_{i(t-kp_i-1)}, \tag{3.3}$$

where p_i – the processing time of task i. This two formulae describe non-singular transformation. On-Off variables model can be obtained by applying expressions 3.2 and 3.3 to Pulse variables model and Step variables model respectively.

Other Time-indexed variable formulations.

An integer Danzig-Wolfe decomposition of constraints can be applied to obtain stronger formulations, i.e. *feasible-subset* formulation suggested in [115] and *chain-decomposition-based* formulation [148]. However weaker or equal formulations are also widely presented in the literature. Comparative analysis with proofs of formulation equivalence is presented in [].

3.6.2 Sequencing/naturaldate formulations

These types of formulations contains at least two types of variables: continuous naturaldate start-time variables S_i and sequencing variables $z_{ij} = \{0, 1\}$ to set a partial order on tasks are defined. $z_{ij} = 1$ is similar to existing precedence relation between tasks i and j, i.e. j can start only after i ends. This formulations arise from disjunctive job-shop problems formulations [149, 106]. Overall number of variables is polynomial ($O(n^2)$), but start-time variables are not binary.

Minimal-Forbidden-Set-Based formulation

This formulation bases on the concept of *minimal forbidden set* (MFS) – a subset of activities which are not able to be processed simultaneously due to the resource capacity constraint violation. In general number of forbidden sets grows exponentially in number of tasks. An algorithm for definin all MFS was proposed in [150]. Let we denote a set of all MFSs by F. Then the model can be formulated as follows.

- Objective function – makespan minimization:

  $$\min S_{n+1},$$

  subject to the following limitations.

- Sequencing variables are binary:

  $$\forall i, j \in N : \; z_{ij} = \{0, 1\}.$$

- There are no sequencing cycles with the length two:

$$\forall i,j \in N, i < j: \; z_{ij} + z_{ji} \leq 1,$$

or more

$$\forall i,j,h \in N, i \neq j \neq h: \; z_{ij} + z_{jh} - z_{ih} \leq 1.$$

- Naturaldate variables satisfy task domains:

$$\forall i \in N: \; r_i \leq S_i \leq D_i - p_i.$$

- Precedence constraints:

$$\forall (i,j) \in E: \; z_{ij} = 1.$$

- All activities of each MFS are not able to be processed simultaneously:

$$\forall F_k \in F: \sum_{i,j \in F_k, i \neq j} z_{ij} \geq 1.$$

- Naturaldate and Sequencing variables are connected by the following constraint:

$$\forall i,j \in N (i \neq j): \; S_j - S_i - (D_j + p_i - p_j - r_i) z_{ij} \geq p_j + r_i - D_j.$$

The Flow-Based formulation

Suggested in [151] the Flow-Based formulation bases on the idea of resource transfer from one task to another. Flow variable ϕ_{ij}^k equals to the amount of resource of resource k which task i transfer to j. This allows to replace forbidden sets constraints by resource flow constraints.

3.6.3 Positional-date/assignment formulations

In this type of formulations (also called event-based) planning horizon is divided by the set of events Ev – possible start and end times of task processing. Event-based formulation was firstly proposed in [152] for machine scheduling problem. RCPSP adaptations are presented in [153, 154, 155]. Two models for different types are presented below.

The Start/End Event-Based formulation

Binary variables x_{ie}^+ and x_{ie}^- equal to 1 if task i starts/ends at event $e \in Ev$. Variables t_e are defined for all events $e \in Ev$ and resource usage variables b_{er} are defined for any pair $e \in Ev, r \in R$. The following model is proposed in the research [37] and corrected by [156].

- Objective function – makespan minimization:

$$\min t_n,$$

  subject to the following constraints.

- Start/End event variables are binary:

$$\forall i \in N, e \in Ev: \ x^+_{ie} = \{0, 1\}, \ x^-_{ie} = \{0, 1\}.$$

- Time and resource variables are non-negative:

$$\forall e \in Ev: \ t_e \geq 0,$$

$$\forall e \in Ev, r \in R: \ b_{er} \geq 0.$$

- First event starts at time 0:

$$t_0 = 0.$$

- Event ids are time-ordered:

$$\forall e \in Ev \setminus \{n\}: \ t_{e+1} - t_e \geq 0.$$

- Task processing time must separate start and end events:

$$\forall i \in N, s, f \in Ev, s < f: \ t_f - t_s - p_i x^+_{is} + p_i(1 - x^-_{if}) \geq 0.$$

- For each task only one start and one end event are defined:

$$\forall i \in N: \ \sum_{e \in Ev} x^+_{ie} = 1,$$

$$\forall i \in N: \ \sum_{e \in Ev} x^-_{ie} = 1.$$

- Start and end events of the same activity should be ordered:

$$\forall i \in N, e \in Ev: \ \sum_{f \in Ev, f \leq e} x^-_{if} + \sum_{s \in Ev, s \geq e} x^+_{is} \leq 1.$$

- Precedence constraints:

$$\forall (i,j) \in E, e \in Ev : \sum_{f \in Ev, f \geq e} x^{-}_{if} + \sum_{s \in Ev, s < e} x^{+}_{js} \leq 1.$$

- Resource constraints:

$$\forall r \in R : \; b_{0r} - \sum_{i \in N} a_{ir} x^{+}_{i0} = 0,$$

$$\forall e \in Ev \setminus \{0\}, r \in R : \; b_{er} - b_{(e-1)r} + \sum_{i \in N} a_{ir}(x^{-}_{ie} - x^{+}_{ie}) = 0,$$

$$\forall e \in Ev, r \in R : \; b_{er} \leq c_r.$$

- Horizon, release time and deadline constraints should be satisfied:

$$\forall i \in N, e \in Ev : \; r_i x^{+}_{ie} \leq t_e \leq (d_i - p_i) x^{+}_{ie} + (d_{n+1} - p_{n+1})(1 - x^{+}_{ie}),$$

$$\forall i \in N, e \in Ev : \; (r_i + p_i) x^{-}_{ie} \leq t_e,$$

$$\forall i \in N, e \in Ev : \; t_e \leq (d_i) x^{-}_{ie} + (d_{n+1} - p_{n+1})(1 - x^{+}_{ie}),$$

$$r_{n+1} \leq t_n.$$

This formulation involves polynomial number of variables and constraints.

The On/Off Event-Based formulation

This formulation was proposed in [37]. The main idea is using On/Off variables, which were discussed earlier instead of Start/End ones. This formulation also less variables, since for each event one variable is considered instead of two. The number of constraints is also polynomial.

3.7 Conclusion

In this section, we recalled the basics of RCPSP and discussed the existing algorithms for calculating lower bounds on makespan as well as existing techniques of constraint propagation and Mixed-Integer Linear Programming. Note that tightening domains is useful for all considered solving methods. In two next sections, we develop a new pseudo-polynominal algorithm for calculating a new lower bound for makespan and new methods for constraint propagation, respectively. These procedure are developed in order to speed up the problem resolution and to achieve high quality solutions in shorter time.

Chapter 4

Lower bounds for RCPSP

Several algorithms for finding a lower bound on the makespan for the Resource Constrained Project Scheduling Problem (RCPSP) were proposed in the literature. However, fast computable lower bounds usually do not provide the best estimations and the methods that obtain better bounds are mainly based on the cooperation between linear and constraint programming and therefore are time-consuming. In this section, a new polynomial algorithm is proposed to find a makespan lower bound for RCPSP with time-dependent resource capacities. Its idea is based on a consecutive evaluation of pairs of resources and their cumulated workload. Using the proposed algorithm, several bounds for the PSPLIB benchmark were improved. The results for industrial applications are also presented where the algorithm could provide good bounds even for very large problem instances. The contributions of this chapter were also presented in the following publications: *D. Arkhipov, O. Battaïa, A. Lazarev. An efficient pseudo-polynomial algorithm for finding a lower bound on makespan for Resource Constrained Project Scheduling Problem, European Journal of Operational Research, 275 (1). 35-44*; *D. Arkhipov, O. Battaïa, A. Lazarev. A new polynomial-time algorithm for calculating upper bounds on resource usage for RCPSP problem. 16th International Conference on Project Management and Scheduling, April, 17-20, Rome, Italy, 2018.*

4.1 Introduction

We consider a generalized statement of RCPSP with a time-dependent resource capacity function defined as follows. There is a set of tasks N and a set of renewable resources R. The amount of resource $X \in R$ which can be used by tasks of set N during time slot $[t, t+1)$ is defined by *capacity function* $c_X(t)$. The statement of a constant resource capacity is a particular case of this formulation. For any task $j \in N$, the following parameters are given: p_j – processing time, a_{jX} – required amount of resource $X \in R$.

Precedence relations between tasks are given by directed acyclic graph $G(N, E)$. If an edge $(i \to j) \in E$ exists, it means that task i must be finished before the starting time of task j.

Further, the following parameters can be calculated for each task taking into account the precedence and resource constraints: r_j – release time, D_j – deadline.

The objective is to find a *schedule* with the lowest makespan i.e. with the shortest project duration as defined by Equation 3.1.

A novel pseudo-polynomial algorithm is developed which extends the relaxation of RCPSP to a Cumulative Scheduling Problem by considering pairs of resources. Our approach uses "time-tabling" techniques to adjust the capacity function of resources first and then it calculates a lower bound on the makespan by evaluating highest possible resource loads for each time slot.

4.2 General approach

The considered decision version of RCPSP is formulated as follows.

Problem 1. *Given set of tasks N, set of resources R and deadline (time horizon) T, does any feasible schedule $\pi \in \Pi(N, R)$ exist with a makespan inferior or equal to T, i.e.*

$$\max_{j \in N} C_j(\pi) \leq T. \tag{4.1}$$

Without loss of generality, it is assumed that the project can be started at time $t = 0$. We introduce two dummy tasks $0, n+1 \in N$ which represent the start and the end of the project, i.e. $r_0 = r_{n+1} = 0$, $p_0 = p_{n+1} = 0$, $D_0 = D_{n+1} = T$ and for any $j \in N \setminus \{0, n+1\}$ precedences $0 \to j$ and $j \to n+1$ exist.

The general scheme of finding a lower bound on the makespan is based on the four following procedures.

Procedure 1. Pre-processing. This procedure updates the release times and deadlines for tasks under condition that the makespan is inferior or equal to T. If during these calculations one of the existing constraints cannot be satisfied, there is no feasible solution with such a bound on the makespan.

Procedure 2. This procedure calculates an upper bound on the resource consumption by set of tasks N during time interval $[0, t+1)$ considering all pairs of resources X, Y. Precedence constraints are replaced by release times and deadlines. In this way, also precedence constraints with time lags can be taken into account.

Procedure 3. Procedure 2 is applied for original precedence graph $G(N, E)$ and the graph with reversed precedence relations $\overline{G}(N, \overline{E})$. The objective is to compare, for any resource $X \in R$, the sum of upper bounds on its amount consumed in intervals $[0, t)$ and

$(t, T]$ with the total amount of resource required for all tasks $\sum_{j \in N} a_{jX} p_j$. If the latter is lower than the former, the considered problem is considered infeasible.

Procedure 4. Finally, the binary search part changes time horizon T and then the calculation is restarted.

In the following, each part of the algorithm is discussed in details.

4.3 Procedure 1: pre-processing

We denote the *length of a longest path from* $i \in N$ to $j \in N$ by P_{ij} if there is a path from i to j in graph $G(N, E)$. The calculation of $P_{ij} \geq 0$ for all pairs of tasks $i, j \in N$ having a path from i to j in G can be done using Dijkstra's algorithm [157].

Let us consider all pairs of tasks $i, j \in N$ such that $P_{ij} \geq 0$ and update release times and deadlines using formulae

$$r_j := \max\{r_j, r_i + P_{ij}\},$$

$$D_i := \min\{D_i, D_j - P_{ij}\}.$$

If for any $j \in N$, holds $D_j - r_j < p_j$, then inequality (4.1) is violated and the algorithm terminates. Otherwise, the compulsory part of the time interval (between the release time and deadline) is calculated for each task $j \in N$ $[CP_j^s, CP_j^e)$ using formulae $CP_j^s = D_j - p_j$, $CP_j^e = r_j + p_j$ (Fig. 4.1). This idea was formulated in [100].

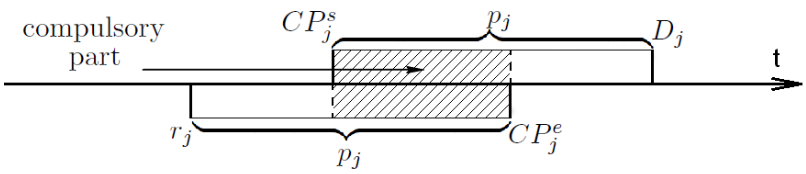

Figure 4.1: Compulsory part of a time interval of task j.

If $CP_j^s < CP_j^e$, then under any schedule π, which satisfies given release dates and deadlines, task j consumes exactly a_{jX} of resource $X \in R$ at each moment of time $t \in [CP_j^s, CP_j^e)$. Therefore, the amount of resource $X \in R$ that can be used by other tasks at each moment of time $t \in [CP_j^s, CP_j^e)$ is not more than $c_X(t) - a_{jX}$. This idea leads us to replace capacity function $c_X(t)$ by function $c'_X(t)$ representing the amount of resource $X \in R$ which can be used to perform non-compulsory parts of tasks (Fig. 4.2)

$$c'_X(t) = c_X(t) - \sum_{j \in N | t \in [CP_j^s, CP_j^e)} a_{jX}.$$

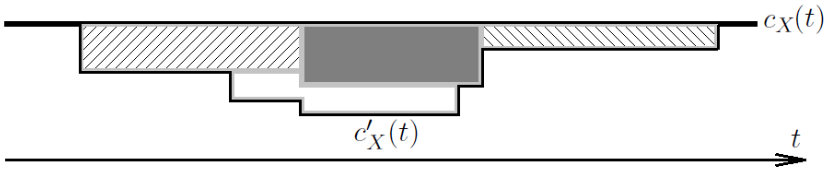

Figure 4.2: Amount of resource $X \in R$ which can be used to perform non-compulsory parts of tasks.

If for any $X \in R$ and $t = 0, \ldots, T-1$ inequality $c'_X(t) < 0$ holds, there is no feasible schedule which satisfies deadline T. The number of breakpoints of function $c'_X(t)$ is not superior to $2n + m$, where m is the number of breakpoints of $c_X(t)$ in horizon T. The complexity of $c'_X(t)$ calculation for all $X \in R$ can be estimated by $O((n+m)r)$ operations, where $n = |N|$ and $r = |R|$. Note that the calculation of $c'_X(t)$ is similar to the Resource profile calculation presented in [129, 130].

The idea of the following algorithm is close to sweep algorithms presented in [131, 132, 133], the difference lies in the utilization of function $c'_X(t)$ which is actively used and dynamically changed in our algorithm.

For each task $j \in N$ and resource $X \in R$, its demand in resource X is compared with the availability of resource X i.e. the value of capacity function $c'_X(t)$ for all m' breakpoints of function $c'_X(t)$ which does not belong to compulsory interval $[CP^s_j, CP^e_j)$. If for any set of breakpoints $t_0, \ldots, t_{m'}$, $c'_X(t) < a_{jX}$ i.e. the amount of resource X is not sufficient to perform task j, the following updates are realized:

- if for any $l \in \{0, \ldots, m'-1\}$ such that $t_l < \max\{CP^s_j, CP^e_j\}$ and $r_j \leq t_{l+1}$ holds $c'_X(t_l) < a_{jX}$, update $r_j := t_{l+1}$;
- if for any $l \in \{1, \ldots, m'\}$ such that $\max\{CP^s_j, CP^e_j\} < t_l$ and $t_{l-1} < D_j$ holds $c'_X(t_{l-1}) < a_{jX}$, update $D_j := t_{l-1}$ (Fig. 4.3).

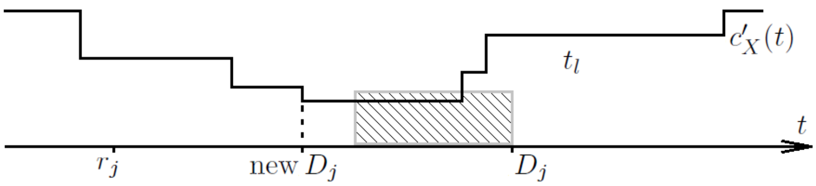

Figure 4.3: Update of deadlines

If for any task $j \in N$, its release date or deadline is updated, the preprocessing part is restarted with the new values of r_j and D_j. Otherwise, the preprocessing algorithm terminates successfully.

Lemma 1. *The complexity of the preprocessing part is $O(n^2(n+m)Tr)$ operations, where n is the number of tasks, m is the highest number of breakpoints of the resource capacity function, T is the time horizon and r is the number of resources.*

Proof. The calculation of P_{ij} for all $i, j \in N$ takes $O(n|E| + n^2 \log n)$ operations, where $|E|$ is the number of edges in graph G. Each iteration of the first round for release and deadline calculation takes $O(n^2)$ operations for checking all paths and $O(n(n+m)r)$ for resource inequalities verification. The number of iterations is no more than nT since each task cannot have more than T release time or deadline updates. Therefore, the total complexity of the preprocessing part can be estimated by $O(n^2(n+m)Tr)$ operations. □

4.4 Inner cycle: relative resource load calculation

Let us consider two resources: X and Y. The earliest possible moment of time when $j \in N$ can start to use resources $X, Y \in R$ is r_j. For any $t \leq p_j$ in time interval $[r_j, r_j+t)$, the amount of resources X and Y consumed by task j cannot be more than $t \cdot a_{jX}$ and $t \cdot a_{jY}$ respectively. If $[CP_j^s, CP_j^e) \neq \emptyset$ task j uses exactly a_{jX} and a_{jY} in each of time slots $[CP_j^s, CP_j^s + 1), \ldots, [CP_j^e - 1, CP_j^e)$. Let

$$A_{jX}(t) = (\min\{t, CP_j^s, CP_j^e\} - \min\{t, r_j - 1\}) \cdot a_{jX}$$

– be the highest possible amount of resource X used by the non-compulsory part of task j in interval $[0, t+1)$ (Fig. 4.4).

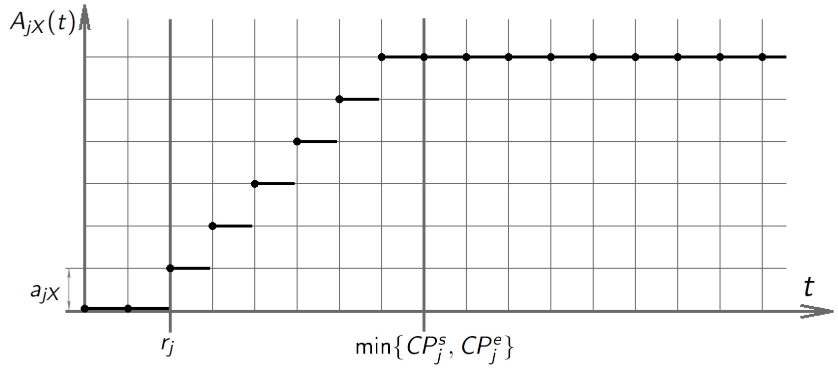

Figure 4.4: $A_{jX}(t)$ – the highest possible amount of resource X used by the non-compulsory part of task j in interval $[0, t+1)$.

The inner cycle procedure processes time slot by time slot starting at the moment $t = 0$. In each time slot, the highest possible consumption of resources X and Y by all tasks of set N is estimated by taking into account only non-compulsory parts of tasks. The amount of resource X used by non-compulsory part of task $j \in N$ in interval $[0, t+1)$

is denoted by $u_{jX}(t)$, and the total consumption of resource X by all tasks in interval $[0, t+1)$ is denoted by $U_X(t) = \sum_{j \in N} u_{jX}(t)$.

The main idea behind the developed algorithm is to calculate an upper bound on the possible consumption of resources X and Y taking into account the non-compulsory parts of the tasks that can be assigned to each time interval. In a general case, the demand in resources of all such tasks will be superior to the resource capacity, but not necessarily in the same proportion for resource X as for resource Y. The originality of the proposed approach is to take into account the fixed proportion of usage of different resources by each task. To calculate the upper bound, a linear combination of fractional parts of such tasks that use the highest available amount of both resources is researched. Among different combinations using the totality of the available resources, a geometric algorithm is used to choose the combination for which the validity of the lower bound on the makespan is proven by Theorems 1 and 2. For example, in a general case, this algorithm will prefer the combination of the tasks using both resources to the combination using the tasks requiring only one resource. This is done in order to provide more flexibility in the resource usage for remaining time intervals.

For resources X and Y, the *consumption scheme* φ is defined when non-compulsory used amounts of resources $u_{jX}(t)$ and $u_{jY}(t)$ are known for any task $j \in N$ and time slot $t = 0, \ldots, T-1$. The consumption scheme is *valid* if for any task $j \in N$ and time slot $t = 0, \ldots, T-1$ the following conditions hold

$$u_{jX}(t) \leq A_{jX}(t),$$

$$u_{jY}(t) \leq A_{jY}(t),$$

$$\frac{u_{jX}(t)}{u_{jY}(t)} = \frac{a_{jX}}{a_{jY}}.$$

The first and the second inequalities are associated with the definitions of $A_{jX}(t)$ and $A_{jY}(t)$ respectively. The last equality is very important, since it requires that the proportion of resources X and Y used by task $j \in N$ remains the same in the considered consumption scheme as in any feasible schedule. Note that each feasible schedule with deadline T possesses valid consumption schemes for all resources.

All time slots $t = 0, \ldots, T-1$ are considered one by one in an iterative way and for each of them, the following optimization problem is solved.

Problem 2. *For each $j \in N$ values $u_{jX}(t-1)$ and $u_{jY}(t-1)$ are given and functions $A_{jX}(t)$, $A_{jX}(t)$ are defined. The objective is to determine $u_{jX}(t) \geq u_{jX}(t-1)$ and $u_{jY}(t) \geq u_{jY}(t-1)$ for all tasks $j \in N$ such that $U_X(t)$ and $U_Y(t)$ reach the highest possible value (since we are interested in an upper bound on resource consumption). The following* *constraints should be taken into account*

$$\frac{u_{jX}(t) - u_{jX}(t-1)}{u_{jY}(t) - u_{jY}(t-1)} = \frac{a_{jX}}{a_{jY}},$$

$$u_{jX}(t) \leq A_{jX}(t),\ u_{jY}(t) \leq A_{jY}(t),$$

$$\sum_{j \in N}(u_{jX}(t) - u_{jX}(t-1)) \leq c'_X(t),$$

$$\sum_{j \in N}(u_{jY}(t) - u_{jY}(t-1)) \leq c'_Y(t).$$

If for any time slot there is more than one solution which satisfy these conditions, the solution will be chosen using the following criterion:

$$\min \sum_{j \in N} \sqrt{(u_{jX}(t) - u_{jX}(t-1))^2 + (u_{jY}(t) - u_{jY}(t-1))^2}. \tag{4.2}$$

The necessity of this criterion is explained by Theorem 1. This problem can be reformulated in terms of vectors.

Problem 3. *For time slot t we have a set of two-dimensional vectors $v_1 = (A_{1X}(t) - u_{1X}(t-1), A_{1Y}(t) - u_{1Y}(t-1)), \dots, v_n = (A_{nX}(t) - u_{nX}(t-1), A_{nY}(t) - u_{nY}(t-1))$ associated with all tasks of set N. The objective is to find a set of coefficients $\{\alpha_1, \dots, \alpha_n\} \in [0,1]$ such that the linear combination*

$$L = \alpha_1 v_1 + \dots + \alpha_n v_n$$

has the highest possible projections on the axes (it corresponds to the highest usage of the resources) and satisfies the inequalities $L_X \leq c'_X(t)$ and $L_Y \leq c'_Y(t)$. If there is more than one solution, choose the one with the lowest sum of the vectors lengths i.e. (it corresponds to criterion 4.2):

$$\min \sum_{j \in N} \alpha_j |v_j|. \tag{4.3}$$

Lemma 2. *Problems 2 and 3 are equivalent.*

Proof. In problem 2 we have to find $u_{jX}(t), u_{jY}(t)$ such that $A_{jX}(t) \geq u_{jX}(t) \geq u_{jX}(t-1)$, $A_{jY}(t) \geq u_{jY}(t) \geq u_{jY}(t-1)$ and

$$\frac{u_{jX}(t) - u_{jX}(t-1)}{u_{jY}(t) - u_{jY}(t-1)} = \frac{a_{jX}}{a_{jY}}.$$

Since values $u_{jX}(t-1)$, $u_{jY}(t-1)$, $A_{jX}(t)$ and $A_{jY}(t)$ are given, each pair of values $u_{jX}(t), u_{jY}(t)$ can be associated with a vector $v_j = (u_{jX}(t) - u_{jX}(t-1),\ u_{jY}(t) - u_{jY}(t-1))$

where $u_{jX}(t) = u_{jX}(t-1) + \alpha_j(A_{jX}(t) - u_{jX}(t-1))$ and $u_{jY}(t) = u_{jY}(t-1) + \alpha_j(A_{jY}(t) - u_{jY}(t-1))$, $\alpha_j \in [0,1]$. Therefore linear combination $L = \alpha_1 v_1 + \cdots + \alpha_n v_n$ has projections

$$L_X = \sum_{j \in N} \alpha_j(u_{jX}(t) - u_{jX}(t-1)) = U_X(t) - U_X(t-1),$$

$$L_Y = \sum_{j \in N} \alpha_j(u_{jY}(t) - u_{jY}(t-1)) = U_Y(t) - U_Y(t-1)$$

on axes OX and OY respectively. Since $U_X(t-1)$ and $U_Y(t-1)$ are fixed, the highest possible values of $U_X(t)$ and $U_Y(t)$ correspond to the highest values of L_X and L_Y (the highest usage of the resources). If there is more than one linear combination which satisfies the above conditions, the second objective (4.3) is applied to choose the solution. Note, that

$$\sum_{j \in N} \alpha_j |v_j| = \sum_{j \in N} \sqrt{(u_{jX}(t) - u_{jX}(t-1))^2 + (u_{jY}(t) - u_{jY}(t-1))^2},$$

hence (4.3) is equivalent to (4.2). □

The following geometric algorithm is designed to solve optimally problem 3.

1. Construct the convex centrally symmetric polygon of possible linear combinations of vectors $v_1, \ldots, v_n$ with coefficients in $[0,1]$ as follows. Let $OV = v_1 + \cdots + v_n$. The upper and lower borders of this polygon are associated with the sequences of vectors placed in descending and ascending orders of tangents of the angle formed with the abscissa axis (Fig. 4.5). Further, it is assumed that these vectors are already sorted in ascending order of tangents.

2. Consider point $C(c'_X(t), c'_Y(t))$. If C is outside the polygon, three following subcases are possible.

   a) C belongs to zone Z1, i.e. $C_X \geq V_X$, $C_Y \geq V_Y$. The procedure returns $\alpha_j = 1$ for each $j \in N$ (Fig. 4.6).

   b) C belongs to zone Z2, i.e. $C_Y < V_Y$ and the projection of C on the axis of ordinates intersects the polygon. The procedure returns a set of coefficients α_j, such as $\sum_{j \in N} \alpha_j v_j$ corresponds to the rightmost intersection of polygon and $Y = c'_Y(t)$ (Fig. 4.7).

   c) C belongs to zone Z3, i.e. $C_X < V_X$ and the projection of C on the axe of abscissa intersects the polygon. The procedure returns a set of coefficients α_j, such as $\sum_{j \in N} \alpha_j v_j$ corresponds to the highest intersection of polygon and $X = c'_X(t)$ (Fig. 4.8).

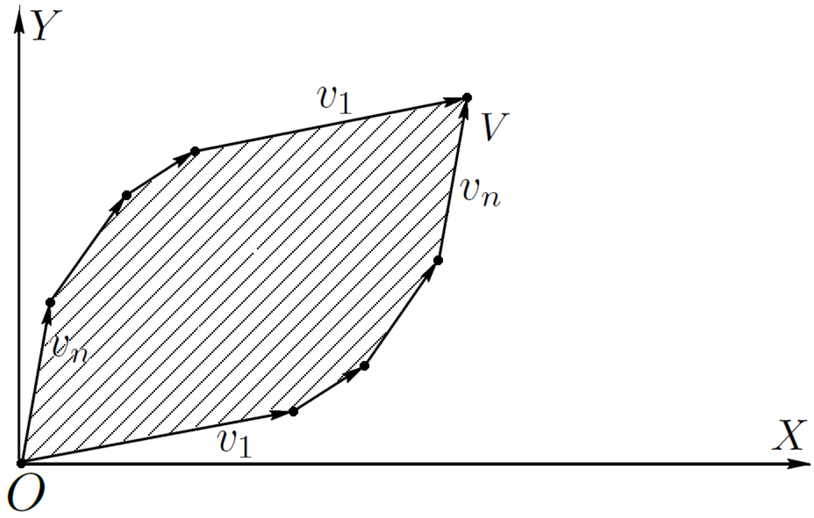

Figure 4.5: Polygon construction.

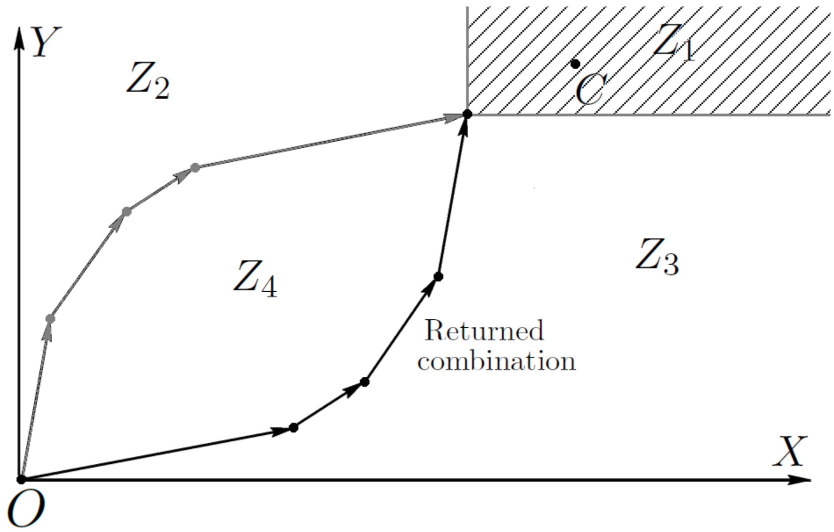

Figure 4.6: Geometric algorithm: subcase 2a.

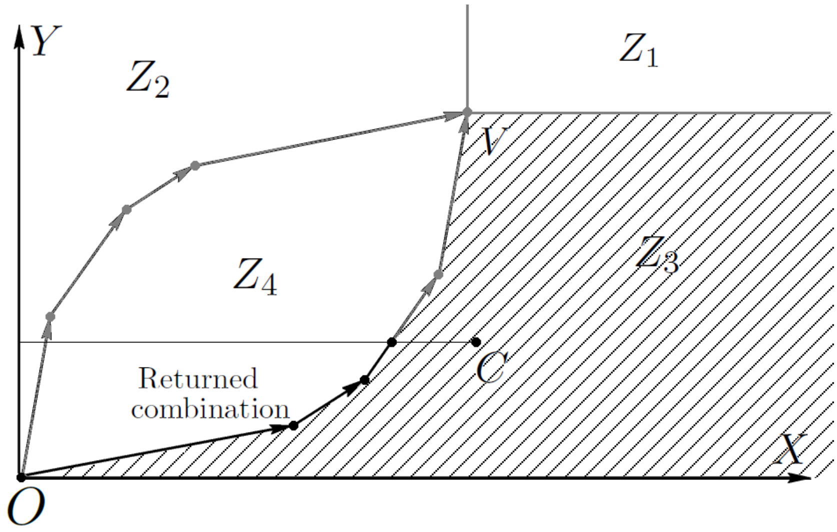

Figure 4.7: Geometric algorithm: subcase 2b.

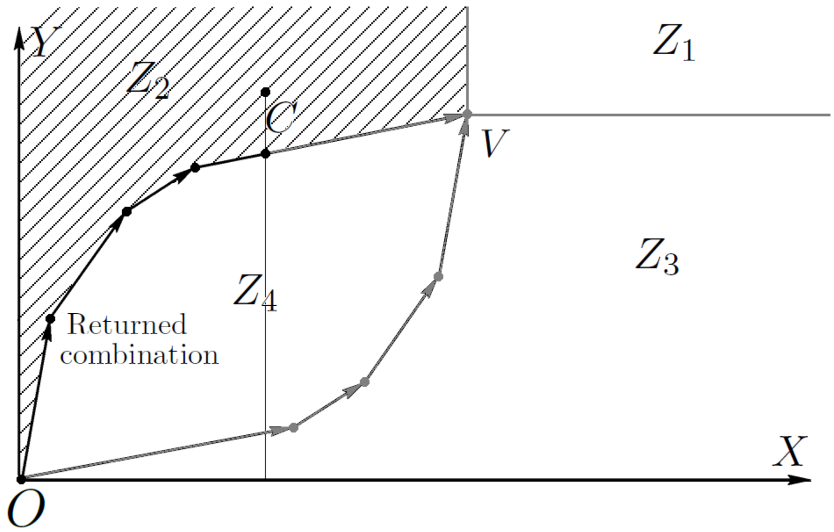

Figure 4.8: Geometric algorithm: subcase 2c.

3. If point C is inside the polygon (zone Z4), we make a translation of the lower border on vector $OC(c'_X(t), c'_Y(t))$ and find the set of coefficients $\{\beta_1, \ldots, \beta_n\}$ which defines the path from point C to V (dashed line in Fig. 4.9), the translated lower border and the borders of the initial polygon, i.e. $\beta_1 v_1 + \cdots + \beta_n v_n = OV - OC$. Then the procedure returns the set of coefficients $\{1 - \beta_1, \ldots, 1 - \beta_n\}$ that corresponds to a polyline which is shown on Fig. 4.9.

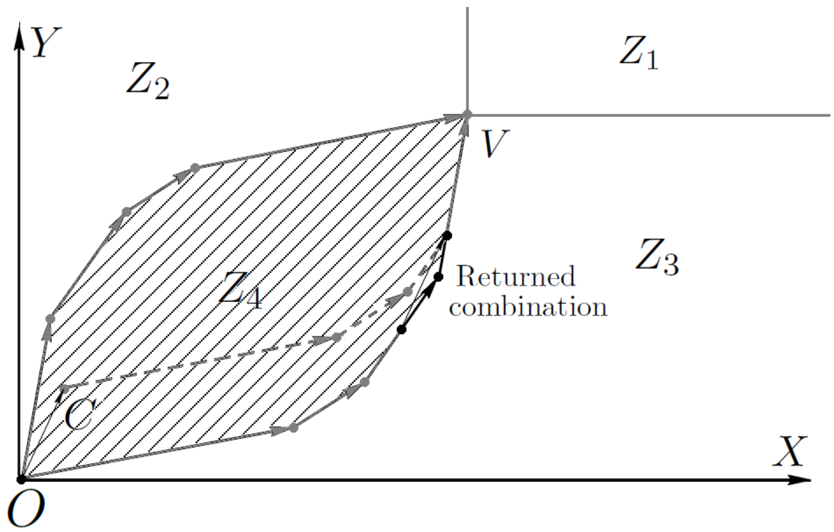

Figure 4.9: Geometric algorithm: step 3 .

The following lemma is required to prove the correctness of the geometric algorithm.

Lemma 3. *Let two sets of vectors $A = \{v_1^A, \ldots, v_l^A\}$ and $B = \{v_1^B, \ldots, v_k^B\}$ such that $\sum_{j=1}^{l} v_j^A = \sum_{j=1}^{k} v_j^B$ and the polygon associated with A be totally included into the polygon associated with B (Fig. 4.10). Then the total length of vectors of set A is not superior to the total length of vectors of set B, i.e.*

$$\sum_{j=1}^{l} |v_j^A| \leq \sum_{j=1}^{k} |v_j^B|.$$

If $A \neq B$, then the inequality is strict.

Proof. Let's compare polygons A and B vector by vector. If we find a difference on vector v_i, let's do an additional construction as on Fig. 4.11, extending vector v_i to the intersection with the polygon B. Then, let us make a centrally symmetric construction for upper border vectors to obtain new polygon B', such as $A \subset B' \subset B$. Then, let us replace polygon B by polygon B'. Obviously, such a change reduces the perimeter. Repeating it no more times than the number of edges of polygon A, we obtain two identical polygons. Q.E.D. □

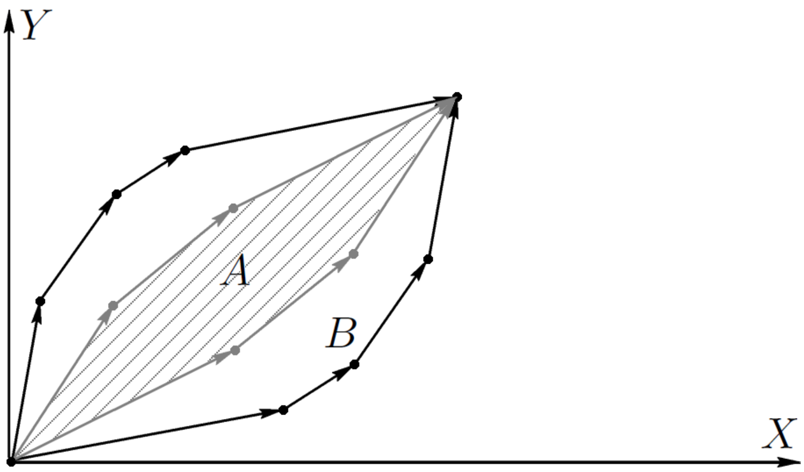

Figure 4.10: Lemma 3. Polygon associated with A be totally included into the polygon associated with B.

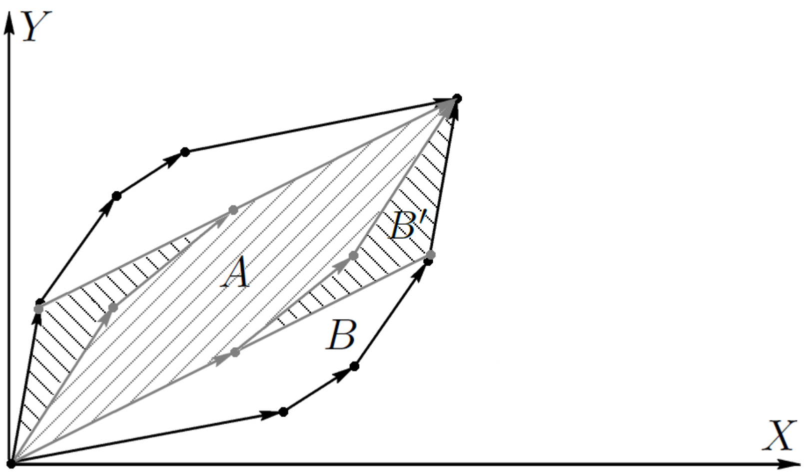

Figure 4.11: Proof of Lemma 3.

Lemma 4. *The proposed geometric algorithm finds an optimal solution for problem 3 in* $O(n^2)$ *operations.*

Proof. Let us show that in each case algorithm finds the set of vectors which sum has the greatest possible projections on the axes (which correspond to the highest possible consumptions of the available resources). If point C is outside the polygon, then the coefficients associated either with the initial set of vectors (2a) or with the intersection of line $Y = C_Y$ with a lower border line of the polygon (2b) or with the intersection of line $X = C_X$ with an upper border line of the polygon (2c) is returned. All this points have the highest possible coordinates among all points of the polygon, which coordinates are not higher than C_X and C_Y. If C lays inside the polygon, then algorithm returns the set of vectors which sum has the coordinates (C_X, C_Y).

Now let us show that the obtained set of vectors $\{\alpha_1 v_1, \ldots, \alpha_n v_n\}$ has the shortest total length among all those with the maximal sum of the coordinates. In the cases where point C is located in zones Z1, Z2 and Z3, there is only one solution which satisfies this condition. When C lies in zone Z4, the algorithm finds the set of vectors $\{\beta_1 v_1, \dots, \beta_n v_n\}$ that corresponds to polyline CV with the longest possible length. Finally, the algorithm returns the set of coefficients $\{\alpha_1, \dots, \alpha_n\} = \{(1-\beta_1), \dots, (1-\beta_n)\}$. Since

$$\sum_{j \in N} |\alpha_j v_j| = \sum_{j \in N} |v_j| - \sum_{j \in N} |\beta_j v_j|,$$

set of vectors $\{\alpha_1 v_1, \dots, \alpha_n v_n\}$ has the shortest possible sum of vector lengths. Therefore obtained solution is optimal with respect to criterion 4.3. Thus Lemma 4 is verified.

The greatest number of operations is required for the case when C lies in zone $Z4$. In this case, a lower border translation is required, the intersection point with the polygon border line can be found in $O(n^2)$ operations. □

The following lemmas should be proved ahead Theorem 1.

Lemma 5. *Let us have a set of two-dimensional vectors $A = \{v_1, \dots, v_m\}$ placed in tangents ascending order and a point V which belongs to the polygon associated with A. Suppose that $A' = \{\alpha_1 v_1, \dots, \alpha_m v_m\}$ is a set of vectors, such that $\forall j = 1, \dots, m : \alpha_j \in [0,1]$, $\sum_{j=1}^{m} \alpha_m v_m = OV$ and the sum of vector lengths $\sum_{j=1}^{m} \alpha_m |v_m|$ is minimal. Then, for any set of vectors $B = \{\beta_1 v_1, \dots, \beta_m v_m\}$, such that $\beta \in [0,1]$ and*

$$\sum_{\beta_j v_j \in B} \beta_j v_j = \sum_{\alpha_j v_j \in A'} \alpha_j v_j = OV,$$

the polygon associated with A' belongs to the polygon associated with B.

Proof. Let us assume the contrary. Suppose that there is a set of vectors B which satisfies the lemma's conditions but the polygon associated with A' does not belong to the polygon associated with B. Lemma 3 implies that the polygon associated with B cannot fully belong to the polygon associated with A'. Therefore, we have to deal only with the situation where the considered polygons are intersected. Hence, polygons' lower border lines have at least four intersection points including O and V. Let's take a look at two consecutive intersection points K and L, such that lower border segment $KL^{A'}$ lies under KL^B. Since both polylines $OV^{A'}$ and OV^B consist of vectors placed in the ascending order of tangents, the vectors which constitute polyline KL^B cannot belong to the set of vectors which constitute $OK^{A'}$ and $LV^{A'}$. Hence, we can replace $KL^{A'}$ by LB^B and thus decrease the perimeter of the polygon associated with A'. This violates the assumption that the sum of vectors lengths of A' is minimal. Lemma 5 is proved. □

Lemma 6. *Suppose that there are two sets of two-dimensional vectors* $A = \{v_1^A, \dots, v_m^A\}$ *and* $B = \{v_1^B, \dots, v_k^B\}$ *such that* $\sum_{j \in A} v_j^A = \sum_{j \in B} v_j^B$ *and the polygon associated with set* A *is totally included in the polygon associated with set* B. *Therefore, there is a set of coefficients* $\alpha_1^1, \dots, \alpha_k^1, \dots, \alpha_1^m, \dots, \alpha_k^m \in [0, 1]$, *which satisfies the following:*

$$\sum_{j=1}^{m} \alpha_1^j = 1,$$
$$\dots$$
$$\sum_{j=1}^{m} \alpha_k^j = 1,$$
$$\sum_{i=1}^{k} \alpha_i^1 v_k^B = v_1^A,$$
$$\dots$$
$$\sum_{i=1}^{k} \alpha_i^m v_k^B = v_m^A.$$

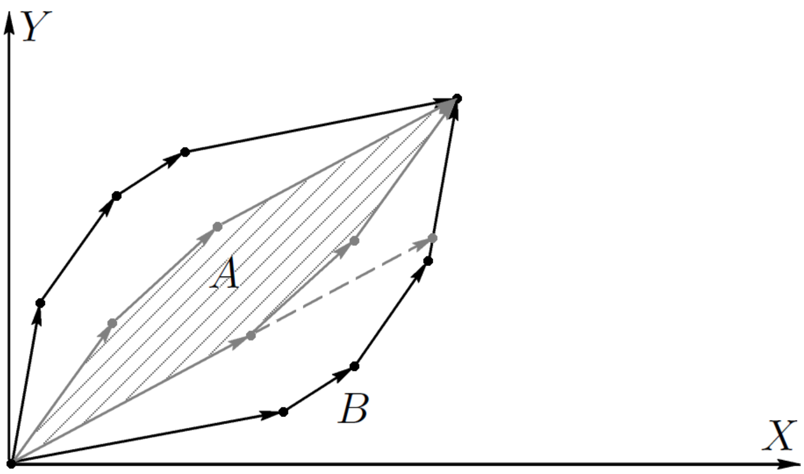

Figure 4.12: Lemma 6.

Proof. Let us find coefficients $\alpha_1^1, \dots, \alpha_k^1$ explicitly, using the graphic approach described in Fig. 4.12 – 4.14. The polygon associated with A is totally included in the polygon associated with B', which is included in the polygon associated with B. Therefore the polygons associated with sets $A' = A \setminus \{v_1^A\}$ and $B' = \{v_1^B - \alpha_1^1 v_1^B, \dots, v_k^B - \alpha_k^1 v_k^B\}$ satisfy the initial conditions of Lemma 6. We can iterate this procedure to find all required sets of coefficients which correspond to all vectors of set A. Q.E.D. □

The presented geometric algorithm considers the time slots one by one in an iterative way. At each step, an optimal solution for problem 2 is found for each pair of resources X

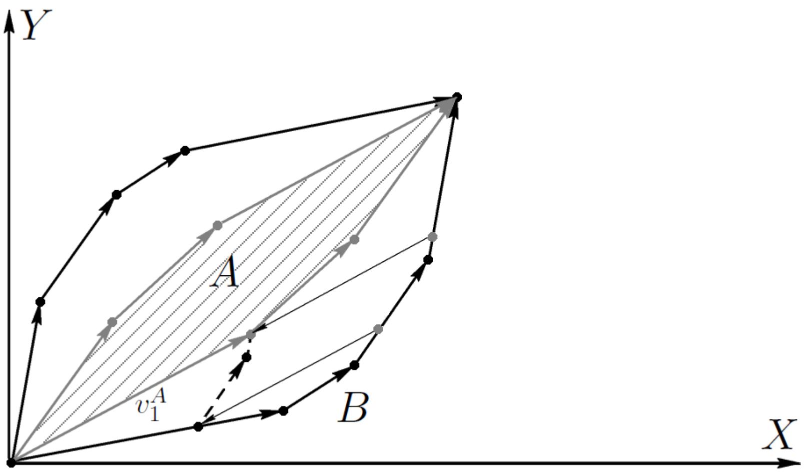

Figure 4.13: Finding $v_1^A = \sum_{i=1}^{k} \alpha_i^1 v_k^B$.

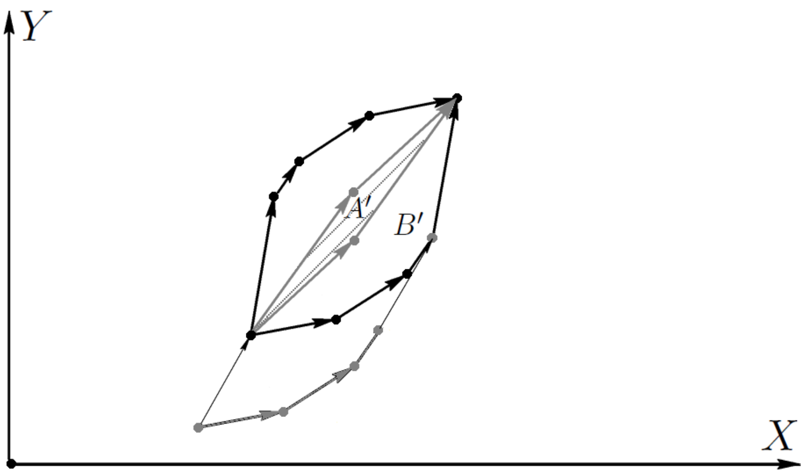

Figure 4.14: New polygons A' and B'.

and Y. Let $U_{X|Y}(t)$ and $U_{Y|X}(t)$ be respectively the amounts of resources X and Y used by set of tasks N in time interval $[0, t+1)$. The following theorem proves that $U_{X|Y}(t)$ and $U_{Y|X}(t)$ provide upper bounds on the consumption of resources X and Y during time interval $[0, t+1)$.

Theorem 1. *Under any valid consumption scheme, the amount of resources X and Y consumed in interval $[0, t+1)$ is not more than*

$$U_{X|Y}(t) + \sum_{t'=0}^{t} (c_X(t') - c'_X(t'))$$

and

$$U_{Y|X}(t) + \sum_{t'=0}^{t} (c_Y(t') - c'_Y(t'))$$

respectively.

Proof. Assume the contrary. Suppose that there is a consumption scheme φ^* which violates the initial assumption and uses more than $U_{X|Y}(t) + \sum_{t'=0}^{t}(c_X(t) - c'_X(t))$ resource X in time interval $[0, t+1)$. If there is more than one of such schemes, consider the one which uses the highest total amount of resources X and Y in time interval $[0, t+1)$. Let $u^*_{jX}(t)$ be the amount of resource X used by task j under consumption scheme φ^* in time interval $[0, t+1)$.

Let us consequently consider the resource consumption at $u^*_{jX}(t')$ for $t' = 0, \ldots, t$. Suppose t' is the first moment of time which satisfies $u^*_{jX}(t') \neq u_{jX}(t')$ or $u^*_{jY}(t') \neq u_{jY}(t')$ for some $j \in N$. We consider polygon OV associated with the set of vectors $v_j = (A_{jX}(t') - u_{jX}(t'-1), A_{jY}(t') - u_{jY}(t'-1))$ corresponding to the consumptions of resources by each task $j \in N$. The vertex of OV with the highest coordinates X and Y is denoted by V. We also consider point $C(c'_X(t), c'_Y(t))$.

There are four possible cases of positioning C in zones Z_1, Z_2, Z_3, Z_4 in relation to polygon OV.

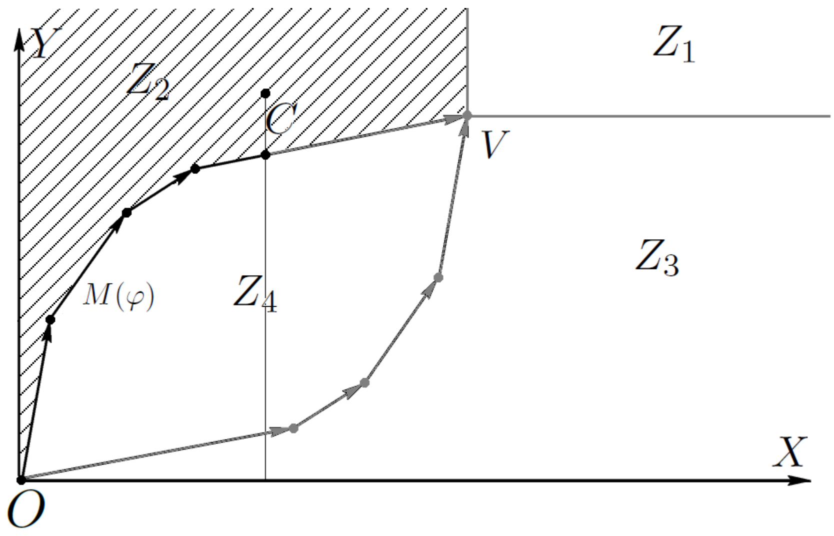

Figure 4.15: The highest possible consumption subject to $C \in Z_2$.

1. $C \in Z_1$. Each resource cannot be totally used, i.e. $(\sum_{j\in N} v_j)_X < c'_X(t')$ and $(\sum_{j\in N} v_j)_Y < c'_Y(t')$. In this case, for any $j \in N$, the following conditions hold $u_{jX}(t') = A_{jX}(t')$ and $u_{jY}(t') = A_{jY}(t')$. Therefore, we can change the consumption during time slot $[t', t'+1)$ for φ^* using the full amounts of resources as well as under φ without violation of any assumption.

2. $C \in Z_2$. Resource Y cannot be totally used. If under φ^* not full amount of resource X is used, we can use it by one of task $j \in N$ which $a_{jX} > 0$ and $u_{jX}(t') < A_{jX}(t')$.

Such a change will not decrease the values of functions $U^*_{jY}(t)$ and $U^*_{jX}(t)$ for any t. Therefore, we can consider only the case where $U^*_{jX}(t') - U^*_{jX}(t'-1) = c'_X(t')$. Note that the highest resource consumption can be achieved only by using a linear combination of vectors with the highest possible ratio $\frac{a_{jY}}{a_{jX}}$ (Fig. 4.15). Hence, if there is a difference in resource consumption between φ and φ^*, then there is a task $j \in N$ such that $u_{jY}(t') > u^*_{jY}(t')$ and there is a task $i \in N$ which holds $u_{iY}(t') < u^*_{iY}(t')$ and $\frac{a_{jY}}{a_{jX}} < \frac{a_{iY}}{a_{iX}}$. This means that we can replace the part of task i used in time slot $[t', t'+1)$ by j under φ^* without violation of any constraint. Such a change will not decrease the values of functions $U^*_{jY}(t)$ and $U^*_{jX}(t)$ for any t. Let us apply the same changes until $u_{jY}(t') = u^*_{jY}(t')$ does not hold for any $j \in N$.

3. $C \in Z_3$. This case is similar to the previous one. We only need to swap resources X and Y in the description of case 2.

4. $C \in Z_4$. This means that under φ the full capacities of both resources can be achieved.

   Suppose that under φ^* full capacities of resources X and Y are achieved in time slot $[t', t'+1)$. According to Lemma 4 we obtain that the polygon related to the resource consumption in $[t', t'+1)$ under φ has the lowest perimeter of all polygons related to the highest consumption of resources X and Y. Lemma 5 implies that it is totally included in the polygon related to φ^*. Therefore, we can change resource consumption under φ^* in time slot $[t', t'+1)$ to the consumption used under φ by taking required parts of $\alpha_j v_j$ from the future timeslots of interval $[t'+1, T)$. Lemma 6 implies that it is possible to replace correctly all parts of vectors $v_1, \ldots, v_n$ used for this procedure by linear combinations of vectors $v^*_1, \ldots, v^*_n$, which were used in φ^* previously. Thus, we can make a change in φ^* without violating the conditions of the Theorem and we obtain equal consumptions of φ^* and φ for time slot $[t', t'+1)$ without increasing or decreasing any amount of resources X and Y being used in any time slot $[t'+1, t'+2), \ldots, [t, t+1)$. For the case where full capacities of resources X and Y are not achieved together in time slot $[t', t'+1)$ under φ^*, the consumption scheme φ^* is modified similarly.

Depending on the case we face in time slot $[t', t'+1)$ we apply the procedure which does not decrease the values of functions $U_X(t)$ or $U_Y(t)$. After having been proceeded with all time slots, we obtain $u_{jX}(t) = u^*_{jX}(t)$ and $u_{jY}(t) = u^*_{jY}(t)$ for any $j \in N$ and $t = 0, \ldots, T-1$. Q.E.D. □

4.5 Main cycle: master algorithm

The master part of our algorithm uses Procedure 2 for $G(N,E)$ and the graph with reversed precedence relations $\overline{G}(N,\overline{E})$ to compare, for any resource $X \in R$, a sum of upper bounds on its possible consumed amount in intervals $[0,t)$ and $(t,T]$ with the total amount of resource required for all tasks $\sum_{j\in N} a_{jX}p_j$. If the latter is lower that the former, the considered problem is considered infeasible for time horizon T.

Then, this verification is made for all moments of times $t = 0,1,2,\dots,T-1$ for all pairs of resources $X, Y \in R$, for which functions $U_{X|Y}(t)$ and $U_{Y|X}(t)$ are calculated. Each feasible schedule defines a valid consumption scheme. Theorem 1 implies that for each resource $X \in R$ and any t, an upper bound of the consumption of resource X by tasks in non-compulsory parts of time interval $[0,t+1)$ under any valid consumption scheme can be estimated by function

$$UB_X(t) = \min_{Y\in R} U_{X|Y}(t).$$

Further, the same procedures, including preprocessing, are applied to set of tasks N but for the graph with reversed precedence relations $\overline{G}(N,\overline{E})$, the values of functions $U'_{X|Y}(t)$ are calculated. As a result, for each resource $X \in R$ we obtain a function

$$UB'_X(t) = \min_{Y\in R} U'_{X|Y}(t)$$

which is an upper bound on the consumption of resource X in non-compulsory parts of time interval $(T-t-1,T]$ for the tasks of set N.

After that, the algorithm repeats the same cycle on all moments of time $t = 1,\dots,T-1$ to check if for any resource $X \in R$ a sum of upper bounds on its available capacity in intervals $[0,t+1)$ and $(t+1,T]$ (Fig. 4.16) is not lower than the sum of the demands in this resource by all tasks, i.e.

$$\sum_{j\in N} a_{jX}p_j \ge UB_X(t) + UB'_X(T-t-2) + \sum_{t=0}^{T-1}(c_X(t) - c'_X(t)).$$

If this condition is violated, then the problem is infeasible for time horizon T.

Theorem 2. *Suppose that the master algorithm was used for set of tasks N, set of resources R and time horizon T. If for any $X \in R$ and $t = 0,\dots,T-1$, the following inequality does not hold*

$$\sum_{j\in N} a_{jX}p_j \le UB_X(t) + UB'_X(T-t-2) + \sum_{t=0}^{T-1}(c_X(t) - c'_X(t)), \tag{4.4}$$

then there is no feasible schedule with makespan inferior or equal to T.

Proof. According to Theorem 1 we obtain that for any feasible schedule $\pi \in \Pi(N, R)$, the amount of resource X used by tasks in time interval $[0, t+1)$ does not exceed $UB_X(t) + \sum_{t'=0}^{t}(c_X(t') - c'_X(t'))$. The amount of resource X used in non-compulsory parts of interval $(t+1, T]$ for tasks does not exceed $UB'_X(T-t-2) + \sum_{t'=t+1}^{T-1}(c_X(t') - c'_X(t'))$. Therefore, taking into account compulsory parts for any feasible schedule π, the amount of resource X used in horizon $[0, T]$ does not exceed

$$UB_X(t) + UB'_X(T-t-2) + \sum_{t=0}^{T-1}(c_X(t) - c'_X(t)).$$

If inequality (4.4) is violated, then for each feasible schedule $\pi \in \Pi(N, R)$ the amount of resource X required for processing all tasks of set N cannot be used during time interval $[0, T]$. This proves the statement of the Theorem. □

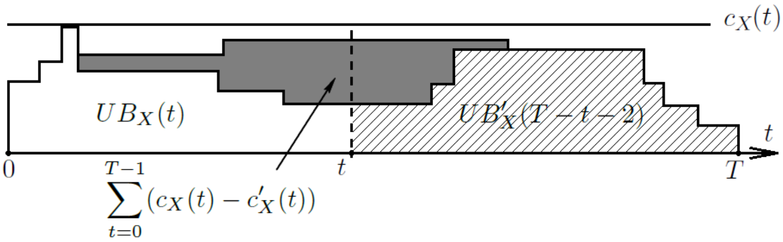

Figure 4.16: Master algorithm.

4.6 Binary search

In this part, a simple binary search is used to find the highest possible value of the time horizon T which satisfies the conditions in Theorem 2.

Theorem 3. *The developed algorithm finds a lower bound on the makespan in $O(n^2 r(n+m+r)T \log T)$ operations, where n is the number of tasks, T is the time horizon, r is the number of resources, m is the highest number of breakpoints of the capacity function of one resource.*

Proof. The number of bi-section search iterations can be estimated by $O(\log T)$ operations. At each iteration, the preprocessing takes $O(n^2(n+m)rT)$ operations. The master part takes $O(n^2 T)$ operations for each pair of resources. Number of pairs of resources is $O(r^2)$. Therefore, the total complexity of the algorithm can be estimated by $O(n^2 r(n+m+r)T \log T)$ operations. □

4.7 Numerical experiments

The algorithm was implemented in C++. Two series of numerical experiments were carried out using Intel Core i7 2.8 GHz CPU with 16 GB RAM. In the first one, the algorithm was tested on the well-known PSPLIB benchmark [39]. In the second one, the algorithm was applied to large-scaled RCPSP instances based on real data provided by Kuznetsov Design Bureau. The results of the tests are presented in Tables 4.1 and 4.2, respectively.

The first series of tests was performed for the problem instances from PSPLIB benchmark. The objective was to compare the results provided by our approach with the best known lower bounds (BKLB), presented at PSPLIB website (consulted in July 2017). The results are given in Tab. 4.1). They show that for 66% of the instances the bound calculated with our algorithm is not worse than the best known lower bound. For all instances, our value of lower bound is not more than $31,5\%$ worse than the best known lower bound, average deviation is about 2%. Moreover, for 4 instances the best known lower bound was improved. It should be noted that the computational time was very short.

Table 4.1: *tasks* – number of tasks, *instances* – number of tested instances, *NW BKLB* – percent of instances, where the obtained lower bound is not worse than the best one, *MIN R BKLB* – minimal ratio of the obtained value to the best known lower bound, *AVG R BKLB* – average ratio of the obtained value to the best known lower bound, *bounds improved* – number of improved bounds, *CPU time* – highest computation time of one instance.

tasks	instances	NW BKLB %	MIN R BKLB %	AVG R BKLB %	bounds improved	CPU time
30	445	66,5	68,5	96,3	0	1 sec
60	450	71,4	77,3	97,6	0	1 sec
90	445	75,3	83,3	98,8	0	2 sec
120	600	54,7	84,7	98,3	4	5 sec

The second series of tests was realized on real industrial data in order to evaluate the possibility to apply the presented approach to real large-scaled instances. The tested instances were created by an arbitrary combination of different projects each of which consisting of 444 tasks and 829 precedence relations. There were 46 different resources required to process each project. Numerical results and a comparison with the project's critical path value are reported in Table 4.2.

Very bad algorithm performance ($LB \approx \frac{C^*_{\max}}{2}$) can be achieved on the instances where under the optimal schedule the resource capacity is poorly used. A simple example of such a "bad instance" is presented in Fig. 4.17. For this instance, the obtained lower

Table 4.2: *tasks* – number of tasks, *precedences* – number of precedence relations, *critical path* – critical path-based lower bound, *obtained LB* – obtained lower bound, *R CP %* – ratio of the obtained lower bound to critical path, *CPU time* – time of algorithm performance

tasks	precedences	critical path	obtained LB	R CP %	CPU time
444	829	830	887	106,9	5m 30s
888	1658	830	1130	136,1	13m 25s
1332	2487	830	1482	178,6	30m 17s
1776	3316	830	1844	222,1	64m 59s
2220	4145	830	2208	266,0	108m 30s

bound is equal to

$$LB = \frac{C^*_{\max}}{2} + 1 = 5,$$

where the optimal makespan value is equal to $C^*_{\max} = 8$.

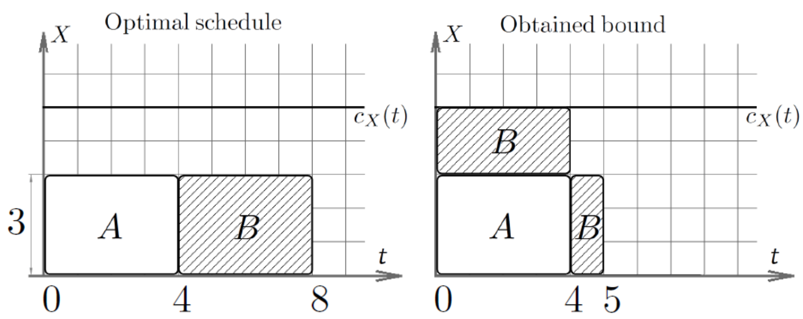

Figure 4.17: Bad instance.

4.8 Generalization for continuous time and piecewise-constant capacity function

We consider the same formulation of RCPSP but with continuous time. There is a set of tasks N and a set of renewable resources R. The capacity of resource $X \in R$ is defined by non-negative piecewise constant function $c_X(t)$ which consists of constant functions $c^1_X(t) = c^1, \forall t \in [0, t_1); c^2_X(t) = c^2, \forall t \in [t_1, t_2); \ldots, c^m_X(t) = c^m, \forall t \in [t_{m-1}, T]$.

We consider the decision version of RCPSP without objective function, but we have to find a schedule which satisfies precedence relations and resource constraints with makespan value lower than T.

For each resource $X \in R$ and any time $t \in [0, T]$ we define an upper bound on highest possible resource consumption in time interval $[0, t)$ by $U_X(t)$. In the previous sections of this chapters we presented a pseudopolynomial algorithm for a discrete version of RCPSP to estimate $U_X(t)$ in $O(n^2r(n+m+r)T\log T)$ operations.

Let us show that the presented algorithm can be used for RCPSP formulation with continuous time. We will call a time point $T_i \in [0,T]$ a *breakpoint* if T_i is a release time of any task $j \in N$, i.e. $T_i = r_j$ or $T_i = t_k$ – start or end of any segment of function $c_X(t)$. Total number of breakpoints is $b \leq n+m$. We assume that the breakpoints are ordered in ascending order: $0 = T_1 < T_2 < \cdots < T_b = T$. Note that the size of a timeslot does not matter for the Master Algorithm. Therefore, we consider intervals $[T_1,T_2),\ldots,[T_{b-1},T_b)$ like timeslots and use Master Algorithm to find functions $u_{jX}(t), u_{jY}(t)$ for each $j \in N$ and each timeslot. According to the theorem 1 obtained functions $U_{X|Y}(t)$ and $U_{Y|X}(t)$ would be an upper bound on resource X and Y consumption respectively for any $t = T_1,\ldots,T_b$ if for any timeslot and any $t \in [T_k, T_{k+1}]$ $u'_{jX}(t) = const$. The following lemma proves that these conditions can be simplified.

Lemma 7. *Suppose there is a set of functions $u_{1X}(t),\ldots,u_{nX}(t)$ defined on timeslot $[T_k,T_{k+1}]$ which satisfy the constraints of the Timeslot problem for any $t \in [0,T]$. Then function*

$$uu_{jX}(t) = \frac{1}{T_{k+1}-T_k}\int_{T_k}^{T_{k+1}} u_{jX}(t)dt$$

defined for all $j \in N$ satisfies the constraints of the Timeslot problem and $uu'_{jX}(t) = const$.

Each feasible schedule defines a set of integrable functions $u_{1X}(t),\ldots,u_{nX}(t)$. Since $uu_{jX}(T_k) \leq u_{jX}(T_k)$, lemma 7 implies that functions $U_{X|Y}(t)$ and $U_{Y|X}(t)$ provided by the Master Algorithm give an upper bound on amount of resources consumed by the tasks belonging to set N in time interval $[0,T_{k+1})$. The complexity of the Geometric Algorithm to solve Timeslot problem is $O(n^2)$. Hence the Master Algorithm complexity equals to $O(n^2(n+m))$, i.e. total number of cycled timeslots (intervals between timepoints) is not more than $n+m$. Therefore functions $U_{X|Y}(t)$ and $U_{Y|X}(t)$ could be found for all pairs of resources $(X,Y) \in R^2$ in $O(r^2n^2(n+m))$ operations, where r – number of resources. An upper bound on resource consumption in interval $[0,t)$ can be defined correctly for any $t = T_1,\ldots,T_n$:

$$U_X(t) = \min_{Y \in R} U_{X|Y}(t).$$

4.9 Conclusion and future perspectives

Novel polynomial algorithms to estimate an upper bound on resource amount used in time interval $[0,t]$ and to find a lower bound on the makespan for RCPSP are presented. This algorithm can be applied to a generalized statement of RCPSP with discrete time where each resource capacity is defined by a non-negative arbitrary function. The main idea of the lower bound calculation is based on a consecutive evaluation of pairs of resources and their cumulated workload. Numerical experiments show that this algorithm provides good results for PSPLIB benchmark instances and it can be used for calculating lower bounds for large-scaled problem instances in reasonable time. An approach can be applied not only to the classic RCPSP formulation but for other RCPSP statements with segment-constant resource capacity functions, i.e. RCPSP/max. Lemma 7 allows decreasing the complexity from $O(n^2 r(n+m+r)T \log T)$ to $O(r^2 n^2 (n+m))$ operations, where n – number of tasks, r – number of resources, m – number of resource capacity function breakpoints and T is the horizon length. This means that lemma 7 changes the algorithm status from pseudo-polynomial to polynomial.

Obtained functions $U_X(t)$ can be used in resource-based propagators to evaluate resource-usage profiles. Our future research will be focused on development of direct methods to improve existing resource-based propagators and to create new techniques of bounding resource usage function.

Chapter 5

Contstraint propagation for RCPSP

5.1 Introduction

Propagators are used as a powerful data pre-processing tool, to make task domains tighter and add new precedence relations. In this chapter, new constraint propagators are proposed. The contributions of this chapter were also presented in the following publication: *D. Arkhipov, O. Battaia, A. Lazarev, G. Tarasov, I. Tarasov. New Task Domain Propagators With Polynomial Complexity For Resource-Constrained Project Scheduling Problem. IX International Conference on Optimization and Applications (OPTIMA2018), October 1-5, Petrovac, Montenegro, 2018.*

We consider a generalized statement of RCPSP, where the capacity of any resource $r \in R$ is defined by some non-negative capacity function $c_r(t)$.

Precedence relations with time lags are given by weighted directed acyclic graph $G = (N, E)$ where E – is the set of edges, defined bu triplets $\{i, j, e_{ij}\}$, where e_{ij} is the time lag between start of tasks $i \in N$ and $j \in N$.

Schedule π defines start times $S_j(\pi)$ for each task $j \in N$. Completion time of any task $j \in N$ under schedule π can be calculated by the formula $C_j(\pi) = S_j(\pi) + p_j$. Schedule π is called feasible, if the following constraints are satisfied.

1. Release times and tails should be satisfied, i.e. for any $j \in N$

$$[S_j(\pi), C_j(\pi)) \subseteq [r_j, T - h_j).$$

2. For any resource $k \in R$ capacity function is not violated for any t, i.e.

$$\sum_{j \in N | t \in [S_j(\pi), C_j(\pi))} a_{jk} \leq c_k(t).$$

3. Precedence relations with time lags should be satisfied. Therefore, any value $e_{ij} \in E$

implies that for start times of tasks i and j the inequality $S_i(\pi) + e_{ij} \leq S_j(\pi)$ holds. Note that values e_{ij} and e_{ji} can both belong to E.

The set of all feasible schedules is defined by $\Pi(N, R)$.

5.2 Data preprocessing

To present new propagation ideas, we need the following notations: $[est_j, lct_j)$ is a processing *domain* for task $j \in N$. In this domain, some time intervals can be forbidden for processing task j.

Note that $ect_j = est_j + p_j$ is the earliest possible completion time and $lst_j = lct_j - p_j$ is the latest possible start time. If $[lst_j, ect_j) \neq \emptyset$, then interval $[lst_j, ect_j)$ is a *compulsory part* of task j.

In this part we make a list of procedures which can improve the efficiency of the majority of the propagators. From the complexity point of view it is more efficient to prepare data by applying procedures 1-5 before using propagators. Procedures 4 and 5 can also be used as other propagators: it is reasonable to execute these procedures if something is changed to get more adjusted results.

1. For any resource $r \in R$, we can create a set of tasks $N_r \subseteq N$ which require this resource, i.e. $N_r = \{j \in N | a_{jr} > 0\}$.

2. The calculation of the longest paths. This takes $O(n|E| + n^2 \log n)$ operations, where $|E|$ – number of edges in graph G, using algorithm presented in [158].

3. Find the earliest/latest starting/completion times using the formulae $est_j = r_j$, $ect_j = r_j + p_j$, $lst_j = T - h_j - p_j$, $lct_j = T - h_j$.

4. If $lst_j < ect_j$ for any task $j \in N$, interval $[lst_j, ect_j)$ sets a compulsory part of j. The complexity of this procedure is $O(1)$ for one task and $O(n)$ for all tasks of the set N. If all compulsory parts of tasks are calculated we can create a set of tasks $N^{CP} \subseteq N$ which compulsory parts are not empty, i.e. $N^{CP} = \{j \in N | lst_j < ect_j\}$. During the process of using domain propagators, est_j and lct_j can be changed and the set N^{CP} can be updated.

5. Calculate the highest possible amount of resource $r \in R$ which can be used by non-compulsory parts of tasks $c_r^n(t)$. This function can be calculated by the formula

$$c_r^n(t) = c_r(t) - \sum_{j \in N: lst_j \leq t \leq ect_j} a_{jr}.$$

For this procedure, the algorithm presented by [130] can be applied with some adjustments for the piecewise constant resource capacity function. The complexity of this procedure is $O(n_r + m)$ operations for single resource where n_r – number of tasks which require resource r during the processing and m – number of breakpoints of $c_r(t)$. If for any task from set N_r, compulsory part $[lst_j, ect_j)$ is updated, it is reasonable to recalculate function $c_r^n(t)$ to obtain a tighter bound on available resources.

5.3 Generalization of existing propagators to piecewise constant capacity function

There is a lot of methods of task domain propagation which are based on resource overload. In this section, we present an adaptation of existing algorithms for the RCPSP generalization with piecewise-constant capacity function and show that in some cases it can improve the efficiency of existing techniques. Presented methods are focused on bounding the earliest starting time, but can be symmetrically used to propagate the latest completion time.

5.3.1 Edge-Finding rule

The idea of Edge-Finding algorithm was firstly proposed in [122]. This method can be divided in two parts. In the first part, specific precedence relations $\Omega \lessdot j$ for any $\Omega \subset N$ and $j \notin \Omega$ are introduced, where $\Omega \lessdot j$ means that under any feasible schedule $\pi \in \Pi(N, R)$, the completion time of j is not earlier than the completion time of all tasks of Ω, i.e.

$$C_j(\pi) \geq \max_{i \in \Omega} C_i(\pi).$$

A fast algorithm to detect all such precedence relations was presented by Vilim [136]. In the second part, for each pair (Ω, j), the following idea is used to tighten the domain of j. Let us consider $[est_\Theta, lct_\Theta)$ – the domain of any $\Theta \subseteq \Omega$, where $est_\Theta = \min_{i\in\Theta} est_i$ and $lct_\Theta = \min_{i\in\Theta} lct_i$. $\Omega \lessdot j$ leads to the fact that all tasks of Θ have to be completed before $C_j(\pi)$ which implies that the amount of resource a_{jr} cannot be consumed by the tasks of Θ in interval $[S_j(\pi), lct_\Theta)$. Let $A_{\Theta r} = \sum_{i\in\Theta} a_{ir}$ be the total amount of resource r required to process Θ. Therefore (Fig. 5.1 a)

$$S_j(\pi) \geq t' = \max_{\Theta\subseteq\Omega}\{est_\Theta + \frac{A_{\Theta r} - (c_r - a_{jr})(lct_\Theta - est_\Theta)}{a_{jr}}\} \tag{5.1}$$

and the earliest starting time can be updated: $est_j := \max\{est_j, t'\}$.

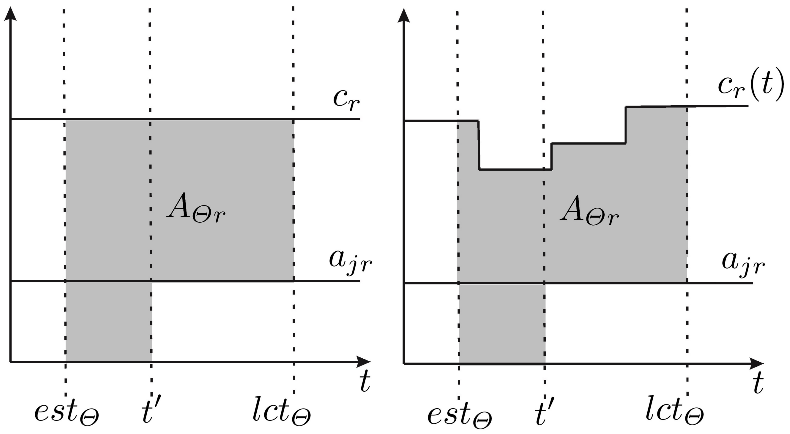

Figure 5.1: Edge-Finding for a) constant resource capacity, b) piecewise constant resource capacity.

For the generalized statement with capacity function, this approach can be used in the same way but with some changes in (5.1) (Fig. 5.1 b).

$$S_j(\pi) \geq t' = \max_{\Theta \subseteq \Omega}\{est_\Theta - \frac{A_{\Theta r} + a_{jr}(lct_\Theta - est_\Theta) + \int_{est_\Theta}^{lct_\Theta} c_r(t)dt}{a_{jr}}\}. \tag{5.2}$$

If the capacity function is piecewise-constant value $\int_{est_\Theta}^{lct_\Theta} c_r(t)dt$ can be calculated in $O(m)$ operations, where m is the number of breakpoints in interval $[est_\Theta, lct_\Theta)$.

5.3.2 Extended Edge-Finding rule

An extended Edge-Finding rule was presented in [138]. It can be formulated as follows. Suppose that task j starts at est_j, $[est_j, ect_j)$ overlaps interval $[est_\Omega, lct_\Omega)$. Then if the total amount of resource required to process j and Ω in interval $[est_\Omega, lct_\Omega)$ is higher than $c_r(lct_\Omega - est_\Omega)$, then $\Omega \lessdot j$ holds, i.e. (Fig. 5.2 a)

$$(est_\Omega \in [est_j, ect_j)) \wedge (A_{\Omega r} + a_{jr}(ect_j - est_\Omega) > C_r(lct_\Omega - est_\Omega)) \Rightarrow \Omega \lessdot j. \tag{5.3}$$

For the generalized statement with capacity function $c_r(t)$, this rule can be written as (Fig. 5.2 b):

$$(est_\Omega \in [est_j, ect_j)) \wedge (A_{\Omega r} + a_{jr}(ect_j - est_\Omega) > \int_{est_\Omega}^{lct_\Omega} c_r(t)dt) \Rightarrow \Omega \lessdot j. \tag{5.4}$$

5.3.3 Time Tabling rule

Time Tabling rule ([130]) is based on the calculation of *resource profile* that is the aggregation of compulsory parts $[lst_i, ect_i)$ of tasks $i \in N$ which holds $lst_i < ect_i$

$$f_r(\Omega, t) = \sum_{i \in \Omega | t \in [lst_i, ect_i)} a_{ir}.$$

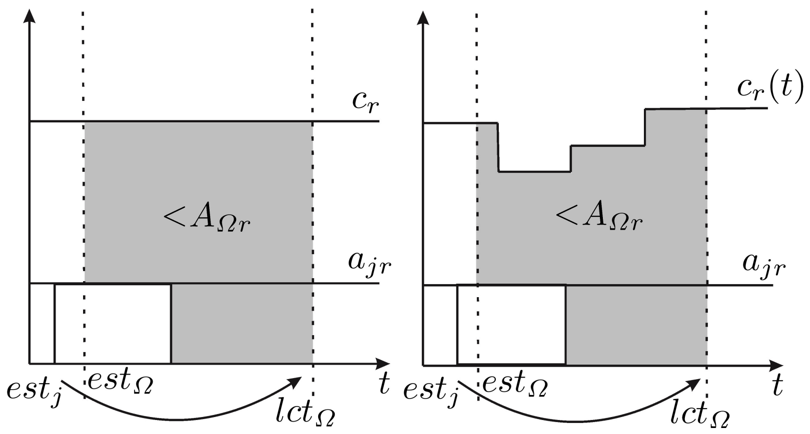

Figure 5.2: Extended-Edge-Finding for a) constant resource capacity, b) piecewise constant resource capacity.

Then the *sweep technique* of [132] is used to check resource overloads (Fig. 5.3 a)

$$(ect_j > t) \wedge (c_r < a_{jr} + f_r(N \setminus j, t)) \Rightarrow est'_j > t.$$

A similar approach can be used to update lct_j. This method was later improved in [133] and [134].

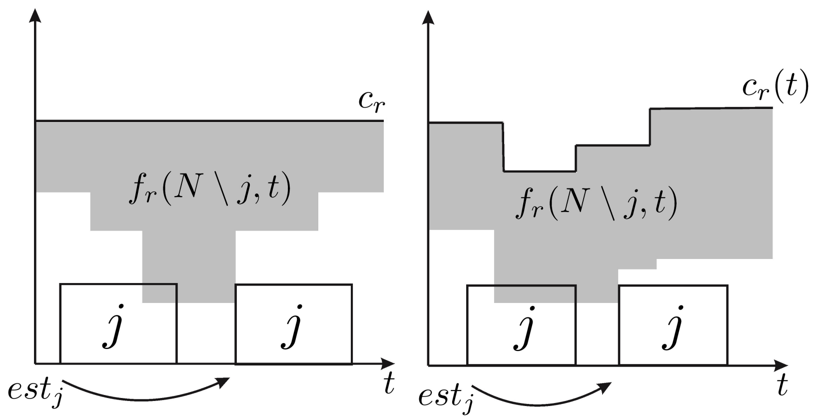

Figure 5.3: Time Tabling for a) constant resource capacity, b) piecewise constant resource capacity.

For the generalized statement, this method can be written as shown in (Fig. 5.3 b)

$$(ect_j > t) \wedge (c_r(t) < a_{jr} + f(N \setminus j, r, t)) \Rightarrow est'_j > t.$$

Note that if in the original algorithm, we need to check overloads only in breakpoints of the function $f_r(N \setminus j, t)$, which belong to interval $[est_j, ect_j)$. In this case, any breakpoint

$t' \in [est_j, ect_j)$ of function $C_r(t)$ has to be checked to get maximal domain propagation of $j \in N$.

5.3.4 Combined Time Table and Extended-Edge-Finding rule

The Extended-Edge-Finding method was enhanced in [139] and [141] by combining it with Time Tabling rule. Let $A^f_{\Omega r}$ be the amount of resource $r \in R$ required to process tasks of set Ω plus amount of resource used by compulsory parts of tasks of set $N \setminus \Omega$ over interval $[est_\Omega, lct_\Omega)$, i.e.

$$A^f_{\Omega r} = A_{\Omega r} + \int_{est_\Omega}^{lct_\Omega} f(N \setminus \Omega, r, t)dt.$$

The substitution of $A_{\Omega r}$ by $A^f_{\Omega r}$ in (5.1) and (5.3) gives Time-Table Extended-Edge-Finding rules.

These rules can be adapted for the generalized statement by substituting $A_{\Omega r}$ by $A^f_{\Omega r}$ in (5.2) and (5.4).

5.3.5 Reinforced Time-Table Edge-Finding rule

The following two rules can be used to make Time-Table Edge-Finding algorithm more efficient.

1. Let us consider $c^n_r(t) = c_r(t) - f_r(N, t)$ that is the highest possible amount of resource which can be used by non-compulsory parts of tasks. In section 5.2, we showed how to calculate this function.

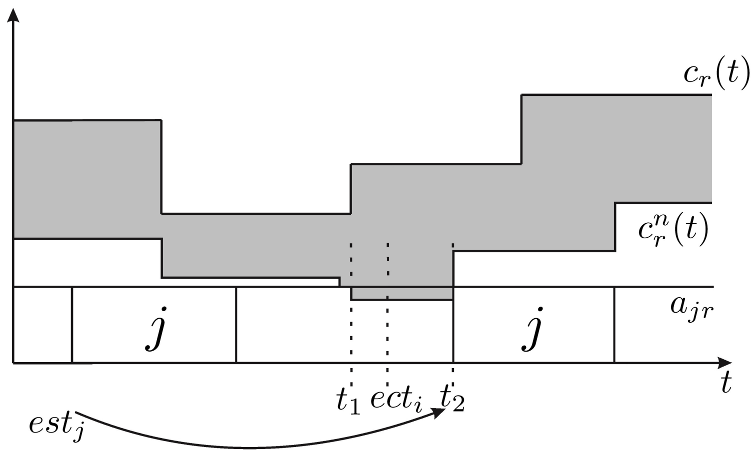

Figure 5.4: Reinforced Edge-Finding rule by considering $c^n_r(t)$ on the interval $[t_1, t_2)$ and a task $i \in \Omega$.

If for any $t \in [t_1, t_2)$ such that $[t_1, t_2) \in [est_\Omega, lct_\Omega) \setminus [ect_j, lst_j)$ holds $c_r^n(t) - a_{jr} < 0$ and there is a task $i \in \Omega$ such that $ect_i \geq t_1$, then the earliest starting time of j can be updated: $est'_j := t_2$ (Fig. 5.4). This enhances Edge-Finding rule for set $\Omega \subset N$ and task $j \in N$ such that $\Omega \lessdot j$.

2. Suppose there is a set $\Theta \subseteq \Omega$ and moment of time $t' \in [est_\Theta, lct_\Theta) \setminus [ect_j, lst_j)$, such that $c_r^n(t') < a_{jr}$ holds. If the amount of the resource required to process non-compulsory parts of all tasks of set Θ is more then $\int_{est_\Theta}^{lct_\Theta} c_r(t) - a_{jr}dt)$, then est_j can be updated, i.e. if the following condition holds, then $est'_j := t'$ (Fig 5.5):

$$A_{\Theta r} - \sum_{i \in \Theta} a_{ir}(|[lst_i, ect_i) \cap [est_\Theta, t')|) > \int_{est_\Theta}^{t'} (c_r(t) - a_{jr})dt$$

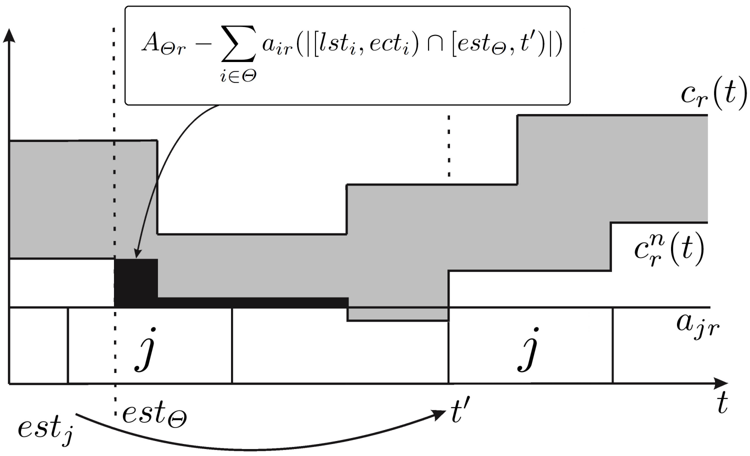

Figure 5.5: Reinforced Edge-Finding rule by considering $c_r^n(t)$ and resource usage in $[est_\Theta, t)$.

These new rules can be combined with the Time-Tabling Edge-Finding technique in order to make it more powerful.

5.4 Adjustment of capacity function

In the previous part, the existing propagators were adapted to the generalized statement with resource capacity function and two new resource-based propagators were proposed. The efficiency of these propagators depends on the amount of resource available during the considered time interval. Therefore, if we consider two instances of problem with the same set of tasks $N^1 \equiv N^2$ with the same precedence relations but with different resource capacities $c_r^1(t)$, $c_r^2(t)$ such that $c_r^1(t) \geq c_r^2(t)$ for any $t \in [0, T)$, the efficiency of domain propagation in the second case will not be worse than in the first one, since task domains will be the same or tighter. In this section, an algorithm to adjust capacity function in order to improve the efficiency of existing resource-based propagators is presented.

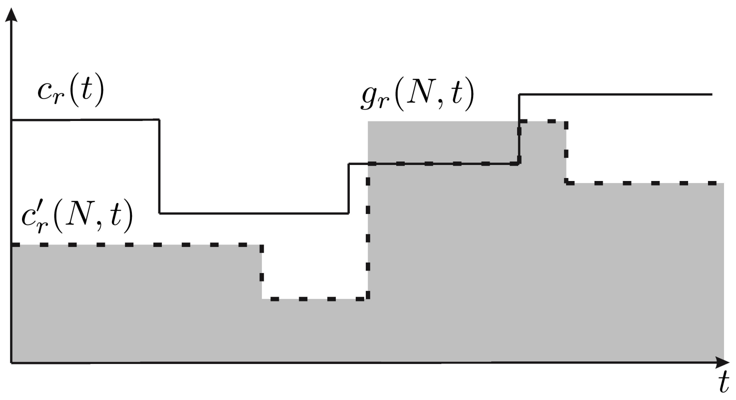

Figure 5.6: Capacity function adjustment.

Let us consider the following function

$$g_r(\Omega, t) = \sum_{j \in \Omega | t \in [est_j, lct_j)} a_{jr}. \tag{5.5}$$

where $g_r(N,t)$ represents an upper bound on the amount of resource which can be consumed by the tasks of N at moment of time t. Therefore, if for any t holds $g_r(N,t) < c_r(t)$, then capacity function $c_r(t)$ can be replaced by $c'_r(N,t)$ i.e. the *highest possible consumption* of resource $r \in R$ by the tasks of the set N (Fig. 5.6). If $c_r(t)$ has a piecewise constant form with m breakpoints, the following algorithm allows calculating $c'_r(N,t)$ in $O(n+m)$ operations.

Note that this algorithm can be also applied to the classic formulation of RCPSP with constant resource capacity.

5.5 Project makespan estimation

All presented methods can be used in order to calculate a lower bound on makespan defined by the following Theorem.

Theorem 4. *If for any time horizon T, after the application of task domain and capacity function propagators one of the following conditions holds, there is no feasible schedule with makespan smaller or equal to T.*

1. *If for any $j \in N$ task domain $[est_j, lct_j)$, is not sufficient to process j, i.e. $lct_\Omega - est_\Omega < \sum_{j \in \Omega} p_j$.*

Data : Set of tasks N, $c_r(t)$ with the set of breakpoints BP_r
Result : $c'_r(N,t)$
$BP_t = \emptyset$;
t set $g_r(N,t) := 0$;
for $j \in N$ **do**
  $\forall t \in [est_j, lct_j)$ increase $g_r(N,t) += a_{jr}$;
  add est_j and lct_j to BP_t;
end
$prev_t = 0$;
$c'_r(N,0) = \min\{c_r(0), g_r(N,0)\}$;
for $current_t \in BP_r \cup BP_t \setminus \{0\}$ *in increasing order* **do**
  $\forall t \in [prev_t, current_t)$ set $c'_r(N,t) := \min\{c_r(prev_t), g_r(N, prev_t)\}$;
  $prev_t := current_t$;
end
Output $c'_r(N,t)$;

Algorithme 1 : The highest possible consumption bound.

2. *Any set $\Omega \subseteq N$ does not have enough resource $r \in R$ to process all tasks of Ω in its domain $[est_\Omega, lct_\Omega)$ i.e. $\int_{est_\Omega}^{lct_\Omega} c_r(t)dt < lct_j - est_j < \sum_{j\in\Omega}(a_{jr}p_j)$.*

3. *If for any resource $r \in R$ and time moment $t \in [0,T)$, the amount of resource $r \in R$ which can be used by non-compulsory parts of tasks is smaller than 0, i.e. $c_r^n(t) < 0$.*

Using logarithmic search in order to find the lowest possible time horizon T^* for which no condition of the Theorem 4 is validated, the lower bound on makespan T^* is obtained. Note that the complexity of this search and the quality of the lower bound depend on the sets Ω. If we consider $\Omega = i \in N$, the complexity is polynomial.

5.6 Forbidden start intervals

Suppose that using compulsory parts with the algorithm suggested by Fox [130] we found a *resource profile* or alternatively we calculated $c_r^n(t)$ i.e. the highest possible amount of resource $r \in R$ which can be used by non-compulsory parts of tasks. Then if for any t in domain of j, such that $t \notin [lst_j, ect_j)$ there is not enough resource $r \in R$ to process task j, i.e. $t \in [est_j, lct_j) \setminus [lst_j, ect_j)$ and $c_r^n(t) < a_{jr}$, then it is not possible to start task j in interval $(t - p_j, t]$ (Fig. 5.7). The set of all such intervals is denoted by P_j, and $P = \cup_{j\in N} P_j$.

Forbidden start intervals can be found by analyzing the resource capacity in the following way. For each resource $r \in R$, we iterate breakpoints t of function $c_r^n(t)$. For all tasks $j \in N$, we check if $t \in [est_j, lct_j) \setminus [lst_j, ect_j)$ and $c_r^n(t) < a_{jr}$, and if it is the case, we add interval $[t - p_j, t')$ into the set of forbidden start intervals for task j, where t' – is the minimum of the next capacity function breakpoint after t and ect_j. We iterate

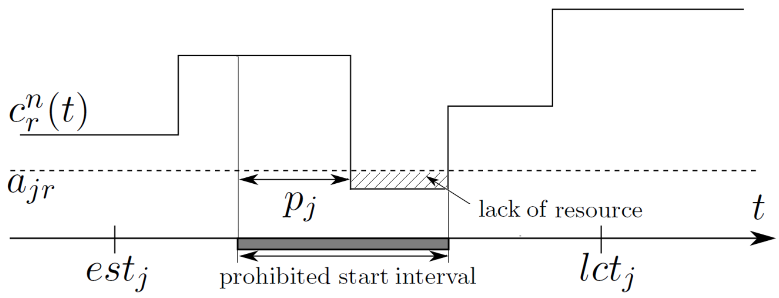

Figure 5.7: The lack of resource r indicates a start interval for task j.

m breakpoints for m resources and for each breakpoint we check $O(1)$ inequalities for n tasks. Therefore, the total complexity of the algorithm is $O(rmn)$.

Note that forbidden start intervals are more informative than task domains. Fig. 5.8 shows an example for task j with a non-empty forbidden start interval while the domain of task is $[0, H)$. It is not possible to start task j in interval $[t_0 - p_j, t'_0)$ but it can be processed at any moment of time $t \in [0, T)$.

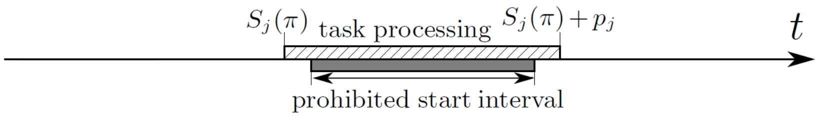

Figure 5.8: There is an interval of forbidden start, but the task can be processed at any moment of time $t \in [0, T)$.

Forbidden start intervals can be also found by analyzing the precedence relations in the following way. Suppose that there is a pair of tasks $(i, j) \in N^2$ with precedence relations e_{ij} and e_{ji}. Then, the set of forbidden start intervals for task j can be enriched using the set of forbidden start intervals for task i. Precedence relations with time lags imply $S_i + e_{ij} \leq S_j$, and $S_j + e_{ji} \leq S_i$, which leads to

$$S_i + e_{ij} \leq S_j \leq S_i - e_{ji}.$$

Then if there is an interval $[t_0, t'_0)$ in which task i cannot be started, then task j cannot be started in interval $(t_0 - e_{ji}, t'_0 + e_{ij})$ as shown in Fig. 5.9.

5.7 Disjunctive functions

The notion of disjunctive graph was defined in [101]. If for a pair of tasks $(i, j) \in N^2$, a disjunctive relation $g_{ij} = 1$ is defined, then tasks i and j cannot be processed simultane-

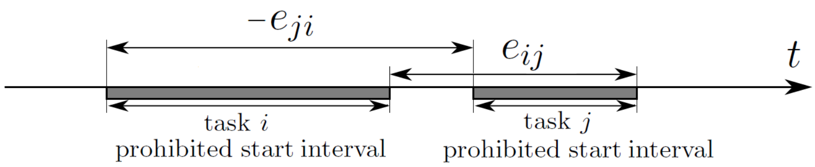

Figure 5.9: Construction of forbidden start intervals using precedences with time lags.

ously. Then, the disjunctive constraint can be formulated as follows. If for a pair of tasks $(i, j) \in N^2$, a disjunctive relation $g_{ij} = 1$ is defined, then processing intervals of these tasks do not intersect, i.e.

$$[S_i(\pi), S_i(\pi) + p_i) \cap [S_j(\pi), S_j(\pi) + p_j) = \emptyset.$$

We use a disjunctive function $g_{ij}(t)$ defined on interval $[0, T)$. If tasks i and j can be processed simultaneously at time t, then $g_{ij}(t) = 0$, otherwise $g_{ij}(t) = 1$. This formulation generalizes the concept of the disjunctive graph where $g_{ij}(t) = 1$ for pair of tasks (i, j).

The following techniques are proposed for calculating disjunctive functions.

1. **Intervals of forbidden processing:** $g_{ij}(t) = 1$ If at time t at least one task i or j cannot be processed.

2. **Resource capacity:** $g_{ij}(t) = 1$ if for any time t and resource $r \in R$ holds

$$a_{ir} + a_{jr} > c_r.$$

3. **Resource capacity with non-compulsory usage function:** $g_{ij}(t) = 1$ if for any $t \notin [lst_i, ect_i) \cup [lst_j, ect_j)$ and resource $r \in R$ the following inequality holds

$$a_{jr} + a_{ir} > c_r^n(t).$$

4. **Input:** New disjunctive functions can be also defined by input.

Now we will show how disjunctive functions can be used for constraint propagation.

Criterion for problem unsolvability showed by disjunctive functions. If for any $(i, j) \in N^2$ such that $g_{ij}(t) = 1$ and $t \in [lst_i, ect_i) \cap [lst_j, ect_j)$, then there is no feasible schedule for this input. Proof: each task has to be processed in its compulsory part, therefore both tasks i, j have to be processed at time t, which contradicts $g_{ij}(t) = 1$.

Generation of new precedence relations using disjunctive functions. Now let us show how disjunctive function can be used for generating new precedence relations. If for two tasks $i, j \in N$ and moment of time t holds $g_{ij}(t) = 1$, and $est_i < t < est_i + p_i$, $lct_j - p_j < t < lct_j$. Then any feasible schedule satisfies the following statements

1. If i starts before t, then task is completed before t.

2. If j is completed after t, then i starts after t.

This means that for tasks i and j, new precedence relation with time lag can be defined (Fig. 5.10)

$$e_{ji} = \min\{est_i - t + p_j, t - lct_j + p_j\}.$$

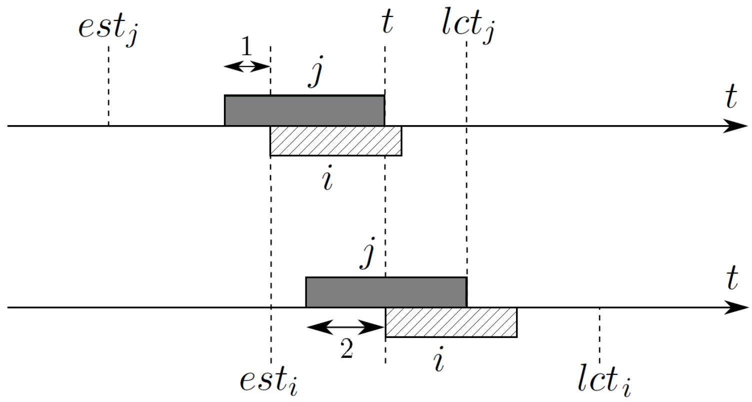

Figure 5.10: Generation of new precedence relations using disjunctive functions. e_{ij} is not less than the minimum of values calculated for cases 1 and 2.

5.8 Conclusion

Constraint programming approach is a very powerful tool for solving scheduling problems. One of the strengths of this method is the possibility of combining different propagators to tighten task domains and to increase the search speed. New propagators based on calculation of forbidden start intervals and disjunctive functions that generalize the concept of disjunctive graphs were presented. In the next sections, the performance of the developed constraint propagators will be evaluated.

Chapter 6

Integration of ergonomic constraints in RCPSP

6.1 Introduction

This chapter addresses the problem of work assignment for operators in aircraft assembly lines (FAL) under economic and ergonomic constraints. We consider a Pulse Aircraft Assembly Line. We suppose that the tasks to be performed at each workstation have been already defined and the set of operators with required skills is already assigned to each workstation. The scope of the optimization problem is one workstation with its tasks and operators. The goal of the considered optimization problem is to assign all tasks to available operators while respecting economic (takt time) and ergonomic constraints. This problem is considered as a special case of Resource Constrained Project Scheduling Problem (RCPSP). A constraint programming (CP) model is developed to solve this problem. The contributions of this chapter were also presented in the following publications: *D. Arkhipov, O. Battaia, A. Lazarev, J. Cegarra. Operator assignment problem in aircraft assembly lines: a new planning approach taking into account economic and ergonomic constraints. Proceedia CIRP, Volume 76, 63-66, 2018*; *D. Arkhipov, O. Battaia, J. Cegarra, A. Lazarev. Work planning in low-volume assembly lines under ergonomic constraints. Proceedia CIRP, Volume 72, 786-789, 2018.*

6.2 Problem definition

The following statement of the problem is considered. There is a set of operators O each of which has only one speciality $s \in S$, where S is the set of specialities. There is a set of resources R, such that for each $x \in R$ the capacity c_x is defined. There is a set of n tasks N to be done. For each task $j \in N$, the following attributes are defined:

- r_j – release time;
- p_j – processing time;
- a_{jx} – amount of resource $x \in R$ required to process task j;
- b_{js} – number of operators with specialty $s \in S$ required to process task j.

For the set of tasks, precedence relations with time lags are defined by a direct weighted graph $G(N, E)$. The existence of edge e_{ji} with weight l_{ji} means that for processing of tasks $j, i \in N$ the following inequality should be satisfied: $S_j(\pi) + l_{ji} \leq S_i(\pi)$, where $S_j(\pi)$ – start time of processing task $j \in N$ under schedule π. Note that values l_{ji} can be negative and both edges e_{ji}, e_{ij} may exist if $l_{ji} + l_{ij} \leq 0$.

Ergonomic impact is measured by a set of methods M. For each triplet $(m \in M, j \in N, s \in S)$, the ergonomic impact of task j on operator with specialty s evaluated by method m is defined by erg_{mjs}. U_{mo} – is the critical level of the total ergonomic impact evaluated by method m for all tasks processed by the same operator o, which should not be violated. Note that this critical level can depend on individual capacities of the operator.

The objective is to find a schedule π^* with the minimal makespan i.e.

$$\min_{\pi} \max_{j \in N} (S_j(\pi) + p_j). \tag{6.1}$$

To solve this problem, we develop a Constraint Programming (CP) and Integer Linear Programming (ILP) approaches.

6.2.1 Constraint programming model

Firstly let we suggest the following constraint programming model to solve formulated problem.

We consider two sets of decision variables.

- $interval_j$ – interval variable associated with the execution of task $j \in N$, i.e. $interval_j = [S_j, C_j)$;
- $assign_{oj}$ – binary variable which is equal to 1 if operator $o \in O$ is assigned to task $j \in N$, otherwise $assign_{oj} = 0$.

The objective is to minimize the makespan as defined by equation 6.1. A solution of the problem must satisfy the following constraints.

The task interval size has to be equal to the task processing time, i.e.

$$\forall j \in N : |interval_j| = p_j. \tag{6.2}$$

For each task $j \in N$, the number of operators with specialty $s \in S$ performing the task has to be equal to b_{js}

$$\forall j \in N,\ s \in S: \sum_{o \in O: s_o = s} assign_{oj} = b_{js}. \tag{6.3}$$

Task processing intervals must satisfy the precedence relations with time lags, i.e.

$$\forall e_{ji} \in E: S_j(\pi) + l_{ji} \leq S_i(\pi). \tag{6.4}$$

The tasks assigned to the same operator cannot overlap, i.e.

$$\forall i, j \in N,\ o \in O: assign_{oi} \cdot assign_{oj} \cdot |interval_i \cap interval_j| = 0. \tag{6.5}$$

The total ergonomic impact of the tasks assigned to the same operator $o \in O$ measured by method $m \in M$ has to be less than the defined critical level U_{mo}, i.e.

$$\forall m \in M, o \in O: U_{mo} \geq \sum_{j \in N} erg_{mjs_o} \cdot assign_{oj}. \tag{6.6}$$

Resource capacity constraint is modelled through a cumulative function which represents the usage of resource $x \in R$ by processing tasks for each time t:

$$F(x,t) = \sum_{j \in N} a_{jx} \cdot f(interval_j, t), \tag{6.7}$$

where $f(interval_j, t) = 1$ if $t \in interval_j$ and $f(interval_j, t) = 0$ otherwise.

Then resource capacity constraint can be formulated as

$$\forall x \in R, t: \ c_x \geq F(x,t). \tag{6.8}$$

6.2.2 Joint CP and ILP model

The second optimization model consists of two parts. The first part uses CP to solve the scheduling problem with the aggregated demand. The objective is to find the schedule with the minimal makespan subject to resource constraints and precedence relations. The second part assigns the operators to the scheduled tasks under the ergonomic constraints. The second part is solved with Mixed-Integer Linear Programming (ILP) approach.

Task Scheduling problem with the aggregated demand

This problem is a relaxation of the initial one because there is no need to assign operators to tasks and specialties are considered as resources. For each specialty, a capacity equals to the number of operators which have to possess this specialty, i.e. we create new resource for each specialty with capacity $c_s = \sum_{j \in N: s_j = s} 1$. If this resource capacity is not violated then under feasible schedule all tasks being processed at moment of time $t \in [0, H)$ require not more than c_s operators with specialty $s \in S$.

There is one set of decision interval variables.

- $interval_j$ – interval variable associated to processing of task $j \in N$, i.e. $interval_j = [S_j, S_j + p_j)$.

The solution of problem must satisfy the following constraints.

- All tasks should be processed in planning horizon

$$H \geq \max_{j \in N}(S_j + p_j).$$

- Interval size should be equal to processing time, i.e. $\forall j \in N$:

$$|interval_j| = p_j.$$

- Task processing interval must satisfy precedence relations with time lags, i.e. $\forall e_{ij} \in E$:

$$S_i + l_{ij} \leq S_j.$$

- Resource capacity constraint modelled through cumulative function which representing usage of resource $x \in R$ by processing task for each time t

$$F(x, t) = \sum_{j \in N} a_{jx} \cdot f(interval_j, t),$$

where $f(interval_j, t) = 1$ if $t \in interval_j$ and $f(interval_j, t) = 0$ otherwise. Then resource capacity constraint can be formulated as

$$\forall x \in R, t : \; c_x \geq F(x, t).$$

The solution of this model provide task processing intervals $[S_j, S_j + p_j)$ which will be used as an input for the following operator assignment problem. The objective is to find a schedule π^* with minimal makespan as introduced by equation 6.1. The solution of this model provides the task processing intervals $[S_j, C_j)$ which will be used as input data for the following operator assignment problem.

Operator assignment problem

To solve this problem we have to assign operators to tasks, which processing intervals $[S_j, C_j)$ are given as a solution of the volume-scheduling problem. Each task $j \in N$ assigned on operator $o \in O$ makes a contribution into total ergonomic impacts erg_{mjs_o} evaluated by method $m \in M$. The objective is to find an assignment with the minimal highest ergonomic impact.

There is one set of binary decision variables.

- $assign_{oj}$ – binary variable equals to 1 if operator $o \in O$ assigned on task $j \in N$, otherwise $assign_{oj} = 0$.

The problem constraints include equations (3) and (6). Since the schedule of the tasks is known, the incompatible sets of tasks can be defined, i.e. the sets of the tasks e that cannot be performed by the same operator. Let E be the family of such incompatible sets, then the constraint on the non intersection of the tasks to be performed by the same operator can be modelled in the following way:

$$\forall e \in E, o \in O : \sum_{j \in e} assign_{oj} \leq 1. \tag{6.9}$$

The objective function is to minimize the highest ergonomic impact calculated for each pair ($m \in M$, $o \in O$).

$$\min \max_{m \in M, o \in O} \sum_{j \in N} assign_{oj} \cdot erg_{mjs_o} \tag{6.10}$$

The solution of this two-part model represents a schedule for tasks with operator assignment.

6.2.3 Numerical experiments

Presented model was implemented on the package IBM ILOG CPLEX. Experimental data was provided by our industrial partner and characterised by 289 tasks with 340 precedence relations, 12 resources, 7 operators with 3 skills and 1 ergonomic evaluation method. Experiments were done using processor Intel(R) Core(TM) i5-4670 3.40GHz and 16 GB of RAM.

The first version of CP model performed well only for a low number of tasks and even for 30 tasks it could not obtain any solution in 1 hour. Joint CP and ILP model showed better results. CP solved volume-scheduling problem for 289 tasks less than in 15 minutes, the second part was solved with ILP in less that 1 second.

6.3 Extension of ergonomic constraints

In addition to the previous model, the following aspects are considered. The set of tasks N is divided into set of groups G and g_j represents the group of task j.

Two types of physical ergonomic constraints for each method $m \in M$ are considered:

- U^h_{mo} – an upper bound on total ergonomic impact evaluated by method m for all tasks processed by operator o in planning horizon H.

- $U^i_{mo}(t_1, t_2)$ – an upper bound on total ergonomic impact evaluated by method m for all tasks processed by operator o in time interval $[t_1, t_2)$, where $t_1, t_2 \in [0, H)$ and $t_1 < t_2$.

To decrease number of mistakes and improve operator's performance of task processing, cognitive load of operators should be taken into account. It is considered that it is more comfortable to work on the same group of tasks as long as possible since a switch from one group to another requires additional information and time for complementary verification. To do it, the minimization of the number of switches between the groups of tasks for operators is defined as the objective function. This means that we have to minimize the number of triplets $(o \in O, i \in N, j \in N)$ such that $g_i \neq g_j$ and under the schedule both tasks are consequently processed by the same operator o.

As previously, this problem can be solved by 2-stages approach using:

1. Constraint programming model to find start time of tasks, with respect to resource, precedence and time horizon constraints. This model remains the same as presented here above.

2. Integer linear programming model (ILP) to assign operators to scheduled tasks.

We also suggest some techniques for data pre-processing in order to increase the speed of solution search.

Operator assignment problem with extended ergonomic constraints

To solve this problem we have to assign operators to tasks, for which processing intervals $[S_j, S_j + p_j)$ are given, or obtained as a solution of the task scheduling problem. The objective is to find a schedule which correctly assigns tasks to operators, does not violate ergonomic physical constraints and minimizes the total cognitive ergonomic score calculated as the number of switches between different groups of tasks for all operators.

Simultaneous processing of two different tasks by the same operator is prohibited. To make this constraint linear, we create a set of pairs of tasks P consisting of ordered pairs of tasks which require the same specialties and have intersected processing intervals, i.e. for each $(i, j) \in P, i < j$ such that $s_i = s_j$ and $[S_i, S_i + p_i) \cap [S_j, S_j + p_j) \neq 0$.

The objective function counts the number of switches between two groups for each operator. If two tasks are executed by the same operator and belong to different groups, then the contribution to the objective function of this triplet of one operator and two tasks equals to 1, otherwise 0. To speed up the solution search, we create a set of all possible ordered pairs of tasks which belong to the same group and can be theoretically processed consequently by the same operator, i.e. for any $(i, j) \in V$ holds

- $g_i = g_j$ – tasks belong to the same group;
- $s_i = s_j$ – tasks require the same specialty;
- $S_i + p_i \leq S_j$ – task i ends before the start of task j.

The following Integer Programming model is developed to solve this problem. There are two sets of decision variables.

- $assign_{oj}$ – binary variable equals to 1 if operator $o \in O$ assigned on task $j \in N$, otherwise $assign_{oj} = 0$.
- seq_{ov} – binary variable equals to 0 if operator $o \in O$ process tasks of the pair $v \in V$ in a sequence, otherwise $seq_{ov} = 1$.

The constraints are formulated as follows.

- Operator can be assigned only to only one task which requires his/her specialty

$$\forall j \in N, o \in O, s_j \neq s_o : \ assign_{oj} = 0.$$

- For each $(i, j) \in P$, operator cannot be assigned to more than one task simultaneously

$$\forall (i, j) \in P, o \in O : \ assign_{oi} + assign_{oj} \leq 1.$$

- Each task $j \in N$ has to receive exact required number of operators

$$\forall j \in N : \ \sum_{o \in O} assign_{oj} = b_j.$$

- For any $o \in O$ and $v = (i, j) \in V$, $seq_{ov} = 1$ if operator o is not assigned to i or j

$$\forall v = (i, j) \in V, o \in O : \ seq_{ov} \geq 1 - assign_{oi},$$

$$\forall v = (i, j) \in V, o \in O : \ seq_{ov} \geq 1 - assign_{oj}.$$

- For any $o \in O$ and $v = (i, j) \in V$, $seq_{ov} = 0$ if operator o does not process any tasks in time interval $[S_i + p_i, S_j]$ $\forall v = (i, j) \in V, o \in O :$

$$seq_{ov} \leq 2 - assign_{oi} - assign_{oj} + \sum_{k \in N : S_i < S_k < S_j} assign_{ok}.$$

- For each task $i \in N$ and operator $o \in O$ there is no more than one pair $(i, j) \in V$ such that i is processed by o just before j

$$\forall i \in N: \ 1 + \sum_{v \in V: v=(i,j)} (seq_{ov} - 1) \geq 0.$$

- For each task $i \in N$ and operator $o \in O$ there is no more than one pair $(j, i) \in V$ such that j is processed by o just before i

$$\forall i \in N: \ 1 + \sum_{v \in V: v=(j,i)} (seq_{ov} - 1) \geq 0.$$

- Total ergonomic score evaluated by method $m \in M$ for operator $o \in O$ is inferior to U^h_{mo}:

$$\forall o \in O, m \in M: \sum_{j \in N} erg_{mj} \cdot p_j \leq U^h_{mo}.$$

There is an additional constraint related to the ergonomic impact on operator o in time interval $[t_1, t_2)$. In the considered industrial context, it is defined by a constant interval length T and a set of upper bounds U^i_m on ergonomic impact evaluated by method $m \in M$ in each interval $[t_1, t_2)$ such as $t_2 - t_1 = T$. This type of constraints should be satisfied for any operator $o \in O$, method $m \in M$ and time interval $[t_1, t_1 + T)$ where $0 \leq t_1 \leq H - T$. This means that feasible assignment of operators should satisfy $|O||M||H - T + 1|$ interval ergonomic constraints. Since time horizon length H can be very large we need some techniques to decrease this number. Let us prove the following lemma.

Lemma 8. *Let $N^s \subseteq N$ be the set of tasks which require specialty s. Then if for any method m, operator o, task $j \in N^{s_o}$ and $t = \{S_j, \max\{0, S_j + p_j - T\}\}$ ergonomic impact on operator o evaluated by method m in time interval $[t, t + T)$ is not more than U^i_m, then all interval ergonomic constraints are not violated.*

Proof 1. *Consider a feasible assignment where operator $o \in O$ is assigned to the set of tasks $N^o \subseteq N^{s_o} \subseteq N$. Then ergonomic impact on operator o evaluated by method $m \in M$ in time interval $[t, t + T)$ is defined by function*

$$Erg_{mo}(t) = \sum_{j \in N^o} erg_{mj} \cdot (\min\{t + T, S_j + p_j\} - \max\{t, S_j\}).$$

Local extremes of this function can only be found when $t + T = S_j + p_j$ or $t = S_j$, i.e. we only need to check that inequality $Erg_{mo}(t) \leq U^i_m$ is correct for $t = S_j + p_j - T$ and $t = S_j$ for each $j \in N^o$. Since $N^o \subseteq N^{s_o}$ the lemma is proved.

Lemma 8 decreases the number of interval ergonomic constraints from $|O||M||H -$

$T+1|$ to $|M| \cdot \sum_{o \in O} 2|N^{s_o}| = 2|N||M||O|$. These constraints can be modelled as follows: $\forall m \in M, o \in O, j \in N$:
$\sum\limits_{i \in N^{s_o}} erg_{mi} \cdot (\min\{S_j + T, S_i + p_i\} - \max\{S_j, S_i\}) \leq U^i_m$,
$\sum\limits_{i \in N^{s_o}} erg_{mi} \cdot (\min\{S_j + p_j, S_i + p_i\} - \max\{S_j + p_j - T, S_i\}) \leq U^i_m$.

The objective is to minimize the total number of switches between the groups of tasks for all operators. For each feasible assignment, the total number of triplets ($o \in O, i \in N, j \in N$) where o processes task i just before j is equal to $|N| - |O|$. The number of pairs of tasks, which belong to different groups consequently processed by operator can be calculated as: *number of pairs of tasks consequently processed by operator and belonging to different groups = total number of tasks processed by operator – number of pairs of tasks consequently processed by operator and belonging to the same group – 1.* Then the objective function can be modelled as follows:

$$\min \sum_{j \in N} b_j - \sum_{o \in O, v \in V} (1 - seq_{ov}) - |O|.$$

6.3.1 Numerical experiments

Presented model was tested on two industrial data instances. Experiments were done on the IBM ILOG CPLEX software using processor Intel(R) Core(TM) i5-4670 3.40GHz and 16 GB of RAM.

The first instance is characterized by 289 tasks with 340 precedences divided into 79 groups, 12 resources, 7 operators with 3 specialities and 1 ergonomic evaluation method. For this instance optimal solution of task scheduling problem was found in 15 seconds. Optimal solution of operator assignment problem was found in 18 seconds.

For the second instance task processing schedule was given and only the solution of operator assignment problem was required. This instance is characterized by 447 tasks divided into 79 groups, 5 operators with 2 specialities and 1 ergonomic evaluation method. For this instance optimal solution was found in 8 hours. We also considered the case without physical ergonomics constraints. This relaxation allowed us to use some additional techniques to improve the speed of search and obtain an optimal solution in 36 seconds.

Another series of numerical experiments was carried out for checking the consistency of preprocessing using developed task domain propagators. Seventy instances were randomly generated based on data of two industrial instances mentioned above. CP model developed for Task Scheduling problem with the aggregated demand was applied to solve this set of instances using IBM CP Optimizer with and without preprocessing. The comparative experiment results are presented in Tab. 6.1. These results show that the use of developed propagators decreases computational time up to 30%. Propagators are especially effective for large-scale instances.

Table 6.1: Comparative experiment results.

№	tasks	horizon	CP	CP + preprocessing	difference
1	20	885	0.0	0.0	0.0
2	20	885	0.0	0.0	0.0
3	20	885	0.0	0.0	0.0
4	20	885	0.0	0.0	0.0
5	20	885	0.1	0.0	-0.1
6	40	2947	0.0	0.1	0.1
7	40	2947	0.0	0.0	0.0
8	40	2947	0.0	0.0	0.0
9	40	2947	0.0	0.0	0.0
10	40	2947	0.0	0.0	0.0
11	60	4904	0.2	0.3	0.1
12	60	4904	0.2	0.2	0.0
13	60	4904	0.2	0.1	0.0
14	60	4904	0.3	0.3	0.0
15	60	4904	0.2	0.2	-0.1
16	80	7026	0.2	1.0	0.8
17	80	7026	0.7	0.5	-0.2
18	80	7026	0.9	0.3	-0.6
19	80	7026	1.0	0.3	-0.7
20	80	7026	0.5	0.9	0.4
21	100	8560	0.5	0.5	0.0
22	100	8560	0.4	0.6	0.2
23	100	8560	0.3	0.4	0.1
24	100	8560	0.6	0.4	-0.2
25	100	8560	0.5	0.6	0.1
26	120	8560	0.6	0.5	-0.1
27	120	8560	0.4	0.5	0.1
28	120	8560	0.6	1.1	0.5
29	120	8560	1.2	0.6	-0.6
30	120	8560	1.3	0.5	-0.7
31	140	8560	1.5	2.2	0.7
32	140	8560	1.7	0.7	-1.0
33	140	8560	1.6	0.7	-1.0
34	140	8560	0.6	4.2	3.6
35	140	8560	3.6	2.1	-1.5
36	160	8560	2.1	3.2	1.1
37	160	8560	0.6	4.4	3.8
38	160	8560	2.1	3.2	1.1
39	160	8560	0.2	0.7	0.5
40	160	8560	2.5	0.9	-1.6

№	tasks	horizon	CP	CP + preprocessing	difference
41	180	8560	1.1	2.6	1.4
42	180	8560	4.8	2.9	-1.8
43	180	8560	5.2	2.7	-2.5
44	180	8560	0.8	2.4	1.7
45	180	8560	2.8	2.4	-0.4
46	200	8830	4.5	2.9	-1.7
47	200	8830	2.2	2.9	0.7
48	200	8830	20.8	2.8	-18.0
49	200	8830	4.3	3.0	-1.2
50	200	8830	5.0	3.4	-1.7
51	220	9158	187.1	141.2	-45.9
52	220	9112	207.7	140.1	-67.6
53	220	9119	215.4	148.1	-67.3
54	220	9106	208.4	154.1	-54.3
55	220	9158	191.9	153.8	-38.1
56	240	9936	224.8	203.2	-21.5
57	240	9908	249.8	227.0	-22.8
58	240	9898	228.6	206.4	-22.2
59	240	9934	210.0	191.0	-18.9
60	240	9903	239.6	191.0	-48.6
61	260	10778	236.4	165.8	-70.6
62	260	10786	236.1	168.9	-67.2
63	260	10776	252.2	167.0	-85.1
64	260	10778	233.8	179.8	-54.0
65	260	10778	250.1	170.6	-79.5
66	280	11702	287.4	196.2	-91.2
67	280	11711	259.9	187.8	-72.1
68	280	11711	303.1	202.9	-100.2
69	280	11729	276.8	222.4	-54.4
70	280	11725	267.7	205.9	-61.8

6.4 Conclusion

In this chapter, we developed new mathematical models in order to integrate ergonomic constraints in RCPSP. These models are suitable to schedule tasks in aircraft assembly lines and can take into account such parameters as labour skills, two types of physical ergonomic constraints evaluated by various ergonomic methods as well as cognitive-oriented parameters here used to reduce the cognitive load of operators. The solution method is based on the two-stage procedure consisting of Constraint programming and Integer Linear programming. Numerical experiments showed that this approach can be used to solve real industrial instances efficiently even for the high numbers of tasks (up to 500). However, since the proposed approach does not evaluate the fairness of workload distribution among the operators, this factor should be studied in future research.

Chapter 7

Workforce planning and scheduling in aircraft assembly lines

7.1 Introduction

Under global trends of hardly predictable demand for customized products and increasing complexity of products, processes and production systems, the human capital is still considered as a source of internal flexibility capable of resolving unpredictable problems. To increase their capacity to recover quickly from disruptions, many enterprises establish cross-training programs in order to form multi-skilled workers. Literature reports that cross-training may improve employee satisfaction, confidence and motivation [46]. Naturally, the organisation of learning programs comes with investments and time and since skill retention is conditioned by practice, managers aim to evaluate the need in multi-skilled workers for the best system performance. However, a recent survey [159] providing a comprehensive overview of skill modelling in workforce planning problems shows that this workforce planning problem has been rarely considered in the literature.

7.2 Constraint programming model

Managers of final assembly lines are interested in improving their planning procedures with the final objective to optimize the efficiency and sustainability of their resources and schedules. The questions to be answered concern both workforce planning perspective and operational scheduling. At the first step, the objective is to define the cross-training program in order to determine the optimal number of multi-skilled workers. At the second step, when the composition of the team is known, the objective is to schedule a large number of tasks under the tact time constraint in the way that working conditions for workers are optimized.

Constraint programming model was created for the considered problem.

Data

The following data is available for optimization:

- T – tact time,
- N – set of tasks to be performed,
- $r_i \in [0, T)$ – release time of task i,
- $d_i \in [r_i, T)$ – deadline of task i,
- p_i – duration of task i,
- e_i – ergonomic penalty for processing task $i \in N$,
- S – set of skills required to perform tasks N,
- $s_i \in S$ – skill required for processing task i,
- $N_s \subseteq N$ – subset of tasks requiring skill s,
- W – set of workers,
- b_i – number of workers required to process task i,
- E_w – total ergonomic constraint defined for worker $w \in W$,
- Z – set of zones of aircraft,
- c_z – capacity of zone z,
- z_i – zone where task i has to be performed,
- $N_z \subseteq N$ – subset of tasks to be performed in zone z,
- Hc – hire cost of one worker,
- Tc – cost of training one worker for one speciality,
- Precedence relations with time lags e_{ij} may be defined for some pairs of tasks $i, j \in N$. The set of precedence relations are defined by P.

Variables

Two sets of variables are considered to find an optimal schedule.

- Set of interval variables $int_i = [start_i, end_i)$ defined for each task $i \in N$.
- $x_{iw} \in 0, 1$ – binary variable, equals to 1 if worker w assigned to task i, and equals to 0 otherwise.

Constraints

Constructed schedule has to satisfy the following constraints

- All given tasks have to be assigned respecting the demand in the number of workers and their skills:
$$\forall i \in N : \sum_{w \in W} x_{iw} = b_i.$$
- The tasks cannot be interrupted and have to be performed by the same workers from the beginning to the end.
- Each worker can perform one single task at time:
$$\forall w \in W, i \neq j \in N : \ (x_{iw} \cdot x_{jw} == 1) \Rightarrow (int_i \cap int_j = \emptyset).$$
- The zone capacity constraint: the number of workers present in each zone of aircraft and at each moment of time is limited, i.e.
$$\forall z \in Z, t \in [0, T) : \sum_{i \in N | t \in int_i} x_{iw} \cdot b_i \leq c_z.$$
- Processing interval constraint:
$$\forall i \in N : \ end_i - start_i = p_i, \ r_i \leq start_i, \ end_i \leq d_i.$$
- Precedence relations with time lags should be satisfied
$$\forall e_{ij} \in P : \ end_i + e_{ij} \leq start_j.$$
- Total ergonomic penalty constraint should be satisfied
$$\forall w \in W : \sum_{i \in N} x_{iw} \cdot e_i \leq E_w.$$

Objective

Objective function evaluates the total cost of the team including single- and multi-skilled workers required for processing all tasks subject to constraints:

$$Hc \cdot \sum_{w \in W} \min\{1, \sum_{i \in N} x_{iw} + Tc \cdot \sum_{w \in W, s \in S} \min\{1, \sum_{i \in N_s} x_{iw}\}.$$

7.3 Numerical experiments

Presented model was implemented using IBM CP Optimizer, processor Intel(R) Core(TM) i5-4670 3.40 GHz, 16 GB RAM. Experimental data was generated based on collected industrial cases, in total dataset includes 85 instances with number of tasks from 10 to 80, number of precedence relations from 10 to 2841, 5 specialties, 10 zones, takt times from 500 to 8000 time units and from 10 to 20 workers available for hiring. Since the number of precedence relations grows with the number of tasks, more instances were generated for a larger number of tasks. In Table 7.1, number of instances generated depending on the number of jobs is presented.

Table 7.1: Number of instances generated for different number of tasks.

number of tasks	number of instances
10	3
20	4
30	7
40	9
50	12
60	14
70	17
80	19

Two series of experiments were carried out. In the first series, the problem instances were solved by the model implemented in IBM CP Optimizer. In the second series, task domains and precedence relations were propagated by developed constraint programming procedures before the solution phase. In Table 7.2, the results of numerical experiments are presented. In total, 41 instances were solved within 30 minutes. The use of developed propagators for these instances allowed to save 21 minutes and 43 seconds which is 5,7% of time spent for solving 41 unpropagated instances. For other 44 instances the gap between obtained objective function and evaluated lower bound was less than 10,3% of objective value. Full results of numerical experiments are presented in the Tab. 7.3, 7.4 and 7.5.

Table 7.2: Numerical experiment results.

instances	time limit	solved	propagated worse	propagated better	equal
85	300 s	17	15	18	52
85	900 s	32	8	10	67
85	1800 s	41	0	0	85

Table 7.3: Experimental results for 300 sec time limit.

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
1	10	33	10	650	0.5	15	0.5	15	0.0	0
2	10	30	10	510	1.6	35	1.5	35	-0.1	0
3	10	40	10	723	0.2	13	0.2	13	0.0	0
4	20	140	10	837	5.4	25	5.6	25	0.2	0
5	20	129	10	1189	35.8	37	31.6	37	-4.2	0
6	20	120	10	1540	18.4	36	20.5	36	2.1	0
7	20	160	20	1170	26.4	37	24.9	37	-1.5	0
8	30	272	10	1111	203.4	38	195.3	38	-8.1	0
9	30	275	10	1561	300.1	111	300.1	111	0.0	0
10	30	264	10	1056	145.2	37	137.7	37	-7.5	0
11	30	253	10	1175	164.4	25	171.5	25	7.0	0
12	30	340	10	1544	29.3	27	34.2	27	5.0	0
13	30	340	20	1106	129.0	37	120.2	37	-8.8	0
14	30	352	20	1690	32.2	25	27.9	25	-4.3	0
15	40	515	10	2012	44.1	16	50.1	16	6.1	0
16	40	498	10	2011	82.2	26	83.7	26	1.6	0
17	40	382	10	2086	300.2	38	300.2	38	0.0	0
18	40	439	10	1489	300.2	40	300.2	40	0.0	0
19	40	400	10	2046	300.2	37	300.2	37	0.0	0
20	40	480	10	1719	300.2	38	300.2	38	0.0	0
21	40	560	10	1544	300.2	38	300.2	38	0.0	0
22	40	640	20	1319	300.2	37	300.2	37	0.0	0
23	40	665	20	1634	300.2	38	300.2	38	0.0	0
24	50	750	10	1600	300.4	38	300.3	38	0.0	0
25	50	702	10	1704	300.4	36	300.4	36	0.0	0
26	50	846	10	1815	300.3	38	300.3	38	0.0	0
27	50	796	10	2115	300.4	37	300.3	37	0.0	0
28	50	826	10	1626	300.4	39	300.4	39	0.0	0
29	50	728	10	1766	300.3	38	300.3	38	0.0	0
30	50	785	10	1847	300.4	38	300.4	38	0.0	0
31	50	800	10	2388	210.7	25	215.5	25	4.8	0

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
32	50	823	10	2388	300.4	39	300.3	39	0.0	0
33	50	1000	10	1805	300.4	38	300.4	38	0.0	0
34	50	980	20	2010	300.3	38	300.4	38	0.0	0
35	50	993	20	2000	300.4	38	300.3	38	0.0	0
36	60	1394	10	3056	300.5	38	300.5	38	0.0	0
37	60	1266	10	3371	301.1	148	301.1	126	0.0	-22
38	60	1361	10	3125	300.6	39	300.6	39	0.0	0
39	60	1173	10	2460	300.5	26	300.5	26	0.0	0
40	60	1095	10	3809	300.6	37	300.6	37	0.0	0
41	60	1329	10	2581	300.5	38	300.5	38	0.0	0
42	60	1194	10	2663	300.5	26	300.5	26	0.0	0
43	60	1389	10	2713	300.6	39	300.5	39	0.0	0
44	60	1276	10	3040	300.6	38	300.5	38	0.0	0
45	60	1368	10	2463	300.6	39	300.5	39	0.0	0
46	60	1337	10	3416	300.6	38	300.5	38	0.0	0
47	60	1440	20	3945	300.5	39	300.5	39	0.0	0
48	60	1414	20	2878	300.5	51	300.5	51	0.0	0
49	60	1478	20	3508	300.5	52	300.6	62	0.0	10
50	70	1693	10	3117	300.8	107	300.8	100	0.0	-7
51	70	1692	10	4046	300.8	73	300.9	108	0.1	35
52	70	1932	10	4698	300.8	105	300.8	107	0.0	2
53	70	1767	10	4720	300.8	109	300.8	100	0.0	-9
54	70	1696	10	4611	300.8	103	300.8	105	0.0	2
55	70	1849	10	3257	300.8	103	300.8	112	0.0	9
56	70	1869	10	4826	300.8	99	300.9	98	0.1	-1
57	70	1928	10	4785	300.8	97	300.9	98	0.1	1
58	70	1676	10	3335	300.8	102	300.9	104	0.1	2
59	70	1742	10	4786	300.8	108	300.8	105	0.0	-3
60	70	1855	10	3611	300.8	100	300.8	105	0.0	5
61	70	1680	10	4012	300.8	98	300.8	100	0.0	2
62	70	1820	10	4528	300.8	49	300.8	49	0.0	0
63	70	1884	10	3262	300.8	39	300.8	39	0.0	0
64	70	1926	20	4855	300.8	62	300.8	104	0.0	42
65	70	1967	20	3774	300.8	38	300.8	38	0.0	0
66	70	1948	20	5308	300.8	109	300.8	49	0.0	-60
67	80	2617	10	4225	301.1	84	301.1	39	0.0	-45
68	80	2603	10	5225	301.3	116	301.1	111	-0.1	-5
69	80	2306	10	4421	301.2	110	301.1	106	-0.1	-4
70	80	2580	10	5158	301.1	110	301.1	120	0.0	10
71	80	2540	10	4185	301.3	107	301.1	63	-0.2	-44
72	80	2567	10	4303	301.1	101	301.1	111	0.0	10
73	80	2424	10	3523	301.2	61	301.1	108	-0.1	47

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
74	80	2408	10	4718	301.2	102	301.1	65	-0.1	-37
75	80	2481	10	4671	301.2	108	301.1	104	-0.1	-4
76	80	2548	10	5367	301.3	51	301.1	111	-0.2	60
77	80	2520	10	4764	301.3	112	301.1	107	-0.2	-5
78	80	2486	10	3811	301.2	112	301.1	104	-0.1	-8
79	80	2554	10	4244	301.2	112	301.1	107	-0.1	-5
80	80	2459	10	4232	301.2	106	301.1	99	-0.1	-7
81	80	2473	10	4103	301.2	108	301.1	51	-0.1	-57
82	80	2649	10	4359	301.1	108	301.1	62	0.0	-46
83	80	2681	20	3400	301.1	38	301.1	38	0.0	0
84	80	2698	20	4147	301.1	105	301.1	106	0.0	1
85	80	2711	20	4538	301.0	38	301.5	38	0.5	0

Table 7.4: Experimental results for 900 sec time limit.

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
					time	obj.	time	obj.	time	obj.
1	10	33	10	650	0.4	15	0.9	15	0.5	0
2	10	30	10	510	1.4	35	1.5	35	0.1	0
3	10	40	10	723	0.2	13	0.2	13	0.0	0
4	20	140	10	837	4.2	25	4.5	25	0.3	0
5	20	129	10	1189	25.4	37	23.1	37	-2.3	0
6	20	120	10	1540	14.6	36	15.6	36	1.0	0
7	20	160	20	1170	18.3	37	17.9	37	-0.4	0
8	30	272	10	1111	143.9	38	142.8	38	-1.1	0
9	30	275	10	1561	522.2	38	514.9	38	-7.4	0
10	30	264	10	1056	101.9	37	101.8	37	-0.1	0
11	30	253	10	1175	676.5	25	676.9	25	0.3	0
12	30	340	10	1544	28.5	27	31.7	27	3.3	0
13	30	340	20	1106	95.3	37	89.0	37	-6.3	0
14	30	352	20	1690	27.2	25	23.6	25	-3.6	0
15	40	515	10	2012	37.9	16	46.1	16	8.2	0
16	40	498	10	2011	76.3	26	82.5	26	6.2	0
17	40	382	10	2086	492.9	38	451.1	38	-41.8	0
18	40	439	10	1489	297.9	40	310.1	40	12.2	0
19	40	400	10	2046	309.0	37	327.5	37	18.4	0
20	40	480	10	1719	558.3	38	512.9	38	-45.4	0
21	40	560	10	1544	408.5	38	385.1	38	-23.4	0

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
22	40	640	20	1319	298.0	37	282.3	37	-15.7	0
23	40	665	20	1634	463.0	38	524.2	38	61.2	0
24	50	750	10	1600	893.2	38	821.5	38	-71.7	0
25	50	702	10	1704	434.5	36	398.7	36	-35.8	0
26	50	846	10	1815	900.3	38	812.5	38	-87.9	0
27	50	796	10	2115	663.6	37	708.3	37	44.7	0
28	50	826	10	1626	900.3	39	900.4	39	0.0	0
29	50	728	10	1766	854.4	38	860.0	38	5.7	0
30	50	785	10	1847	854.0	38	900.3	38	46.3	0
31	50	800	10	2388	177.0	25	175.0	25	-2.1	0
32	50	823	10	2388	900.3	39	900.3	39	0.0	0
33	50	1000	10	1805	900.3	38	900.3	38	0.0	0
34	50	980	20	2010	900.3	38	900.3	38	0.0	0
35	50	993	20	2000	900.3	38	900.3	38	0.0	0
36	60	1394	10	3056	900.5	38	900.5	38	0.0	0
37	60	1266	10	3371	901.0	38	901.0	38	0.0	0
38	60	1361	10	3125	900.5	39	900.5	39	0.0	0
39	60	1173	10	2460	343.6	26	369.8	26	26.3	0
40	60	1095	10	3809	900.5	37	900.5	37	0.0	0
41	60	1329	10	2581	900.5	38	900.5	38	0.0	0
42	60	1194	10	2663	404.7	26	435.9	26	31.2	0
43	60	1389	10	2713	900.5	39	900.5	39	0.0	0
44	60	1276	10	3040	900.5	38	900.5	38	0.0	0
45	60	1368	10	2463	900.5	39	900.5	39	0.0	0
46	60	1337	10	3416	900.5	38	900.5	38	0.0	0
47	60	1440	20	3945	900.5	39	900.5	39	0.0	0
48	60	1414	20	2878	900.5	38	900.5	38	0.0	0
49	60	1478	20	3508	900.5	39	900.5	39	0.0	0
50	70	1693	10	3117	900.7	38	900.7	38	0.0	0
51	70	1692	10	4046	900.8	38	900.7	38	0.0	0
52	70	1932	10	4698	900.7	38	900.7	38	0.0	0
53	70	1767	10	4720	900.7	38	900.7	38	0.0	0
54	70	1696	10	4611	900.7	38	900.7	38	0.0	0
55	70	1849	10	3257	900.7	39	900.7	39	0.0	0
56	70	1869	10	4826	900.7	38	900.7	38	0.0	0
57	70	1928	10	4785	900.7	38	900.7	38	0.0	0
58	70	1676	10	3335	900.7	38	900.7	38	0.0	0
59	70	1742	10	4786	900.7	38	900.7	38	0.0	0
60	70	1855	10	3611	900.7	38	900.7	38	0.0	0
61	70	1680	10	4012	900.7	38	900.7	38	0.0	0
62	70	1820	10	4528	900.7	39	900.7	37	0.0	-2
63	70	1884	10	3262	900.7	38	900.7	38	0.0	0

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
64	70	1926	20	4855	900.7	52	900.7	74	0.0	22
65	70	1967	20	3774	900.7	38	900.7	38	0.0	0
66	70	1948	20	5308	900.7	38	900.7	38	0.0	0
67	80	2617	10	4225	901.0	109	900.9	49	0.0	-60
68	80	2603	10	5225	900.9	114	901.0	114	0.0	0
69	80	2306	10	4421	900.9	56	900.9	43	0.0	-13
70	80	2580	10	5158	900.9	111	901.0	115	0.0	4
71	80	2540	10	4185	900.9	107	900.9	52	0.0	-55
72	80	2567	10	4303	900.9	38	901.0	75	0.0	37
73	80	2424	10	3523	900.9	50	900.9	90	0.0	40
74	80	2408	10	4718	901.0	106	900.9	65	0.0	-41
75	80	2481	10	4671	901.0	108	901.0	110	0.0	2
76	80	2548	10	5367	901.0	51	901.0	111	0.0	60
77	80	2520	10	4764	901.0	115	901.0	111	0.0	-4
78	80	2486	10	3811	901.0	82	900.9	72	0.0	-10
79	80	2554	10	4244	900.9	50	901.0	107	0.0	57
80	80	2459	10	4232	900.9	105	901.1	50	0.1	-55
81	80	2473	10	4103	901.0	111	900.9	83	0.0	-28
82	80	2649	10	4359	901.2	61	901.0	99	-0.2	38
83	80	2681	20	3400	901.5	38	901.0	38	-0.6	0
84	80	2698	20	4147	901.3	62	901.0	61	-0.3	-1
85	80	2711	20	4538	901.0	38	901.0	38	0.0	0

Table 7.5: Experimental results for 1800 sec time limit.

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
1	10	33	10	650	1.1	15	1.0	15	-0.1	0
2	10	30	10	510	1.6	35	1.5	35	0.0	0
3	10	40	10	723	0.2	13	0.2	13	0.0	0
4	20	140	10	837	4.5	25	4.7	25	0.3	0
5	20	129	10	1189	28.9	37	23.7	37	-5.1	0
6	20	120	10	1540	16.1	36	15.3	36	-0.8	0
7	20	160	20	1170	19.4	37	18.7	37	-0.7	0
8	30	272	10	1111	169.9	38	140.0	38	-29.9	0
9	30	275	10	1561	553.5	38	510.3	38	-43.2	0
10	30	264	10	1056	107.4	37	100.5	37	-6.9	0
11	30	253	10	1175	697.2	25	675.5	25	-21.6	0
12	30	340	10	1544	30.8	27	31.2	27	0.3	0

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
13	30	340	20	1106	144.2	37	89.1	37	-55.1	0
14	30	352	20	1690	32.2	25	23.7	25	-8.5	0
15	40	515	10	2012	49.1	16	46.0	16	-3.0	0
16	40	498	10	2011	108.5	26	80.9	26	-27.6	0
17	40	382	10	2086	630.6	38	458.3	38	-172.3	0
18	40	439	10	1489	338.3	40	313.4	40	-24.9	0
19	40	400	10	2046	359.5	37	329.5	37	-30.0	0
20	40	480	10	1719	639.4	38	520.1	38	-119.3	0
21	40	560	10	1544	458.0	38	385.9	38	-72.1	0
22	40	640	20	1319	320.1	37	286.0	37	-34.1	0
23	40	665	20	1634	542.2	38	524.1	38	-18.1	0
24	50	750	10	1600	994.7	38	822.7	38	-172.0	0
25	50	702	10	1704	498.8	36	401.6	36	-97.2	0
26	50	846	10	1815	959.3	38	812.3	38	-147.0	0
27	50	796	10	2115	676.0	37	702.7	37	26.7	0
28	50	826	10	1626	1428.8	39	1777.8	39	349.0	0
29	50	728	10	1766	1081.5	38	866.4	38	-215.1	0
30	50	785	10	1847	899.0	38	927.5	38	28.5	0
31	50	800	10	2388	179.1	25	174.6	25	-4.5	0
32	50	823	10	2388	1414.0	39	1578.6	39	164.7	0
33	50	1000	10	1805	1252.0	38	1195.3	38	-56.7	0
34	50	980	20	2010	1220.9	38	1052.2	38	-168.8	0
35	50	993	20	2000	1280.7	38	1104.3	38	-176.3	0
36	60	1394	10	3056	1800.5	38	1800.5	38	0.0	0
37	60	1266	10	3371	1801.0	38	1801.0	38	0.0	0
38	60	1361	10	3125	1800.5	39	1800.5	39	0.0	0
39	60	1173	10	2460	368.7	26	370.4	26	1.7	0
40	60	1095	10	3809	1349.7	37	1481.5	37	131.9	0
41	60	1329	10	2581	1800.5	38	1800.5	38	0.0	0
42	60	1194	10	2663	413.6	26	447.6	26	33.9	0
43	60	1389	10	2713	1542.4	39	1368.3	39	-174.2	0
44	60	1276	10	3040	1765.9	38	1786.5	38	20.7	0
45	60	1368	10	2463	1380.7	39	1205.6	39	-175.1	0
46	60	1337	10	3416	1800.5	38	1800.5	38	0.0	0
47	60	1440	20	3945	1800.5	39	1800.5	39	0.0	0
48	60	1414	20	2878	1800.5	38	1800.5	38	0.0	0
49	60	1478	20	3508	1800.6	39	1800.5	39	-0.1	0
50	70	1693	10	3117	1800.8	38	1800.7	38	-0.1	0
51	70	1692	10	4046	1800.8	38	1800.7	38	0.0	0
52	70	1932	10	4698	1800.7	38	1800.8	38	0.1	0
53	70	1767	10	4720	1800.7	38	1800.7	38	0.0	0
54	70	1696	10	4611	1800.7	38	1800.7	38	0.0	0

№	tasks	prec. numb.	workers	horizon	CP		CP + pre-processing		difference	
					time	obj.	time	obj.	time	obj.
55	70	1849	10	3257	1800.7	39	1800.7	39	0.0	0
56	70	1869	10	4826	1800.7	38	1800.7	38	0.0	0
57	70	1928	10	4785	1800.7	38	1800.7	38	0.0	0
58	70	1676	10	3335	1800.7	38	1800.7	38	0.0	0
59	70	1742	10	4786	1800.7	38	1800.7	38	0.0	0
60	70	1855	10	3611	1800.7	38	1800.7	38	0.0	0
61	70	1680	10	4012	1800.7	38	1800.7	38	0.0	0
62	70	1820	10	4528	1800.7	37	1800.7	37	0.0	0
63	70	1884	10	3262	1800.7	38	1800.7	38	0.0	0
64	70	1926	20	4855	1800.7	39	1800.7	39	0.0	0
65	70	1967	20	3774	1800.8	38	1800.7	38	0.0	0
66	70	1948	20	5308	1800.7	38	1800.8	38	0.0	0
67	80	2617	10	4225	1801.0	38	1800.9	38	0.0	0
68	80	2603	10	5225	1801.0	38	1801.0	38	0.0	0
69	80	2306	10	4421	1801.0	37	1801.0	37	0.0	0
70	80	2580	10	5158	1801.0	39	1801.0	39	0.0	0
71	80	2540	10	4185	1801.0	38	1801.0	38	0.0	0
72	80	2567	10	4303	1801.1	38	1801.0	38	-0.2	0
73	80	2424	10	3523	1801.1	38	1801.0	38	-0.2	0
74	80	2408	10	4718	1801.0	39	1801.0	39	0.0	0
75	80	2481	10	4671	1801.0	39	1801.0	39	0.0	0
76	80	2548	10	5367	1801.0	38	1801.0	38	0.0	0
77	80	2520	10	4764	1801.0	38	1801.0	38	0.0	0
78	80	2486	10	3811	1801.0	39	1801.1	39	0.1	0
79	80	2554	10	4244	1801.0	39	1801.0	39	-0.1	0
80	80	2459	10	4232	1801.0	38	1801.0	38	0.0	0
81	80	2473	10	4103	1801.0	39	1801.0	39	-0.1	0
82	80	2649	10	4359	1801.1	39	1801.0	39	0.0	0
83	80	2681	20	3400	1801.0	38	1800.9	38	0.0	0
84	80	2698	20	4147	1801.0	38	1801.0	38	0.0	0
85	80	2711	20	4538	1801.0	38	1801.0	38	0.0	0

7.4 Conclusion

In this chapter, a new mathematical model was developed for the problem of workforce planning and scheduling in aircraft assembly lines. The developed planning approach was applied to a dataset of problem instances based on industrial data and showed promising results in terms of solution time and objectives attained. Numerical experiments showed that developed model allows to find optimal and suboptimal solutions of large-scaled instances in reasonable time. Moreover, the preprocessing procedures based on constraint propagators developed in Chapter 5 were used to improve the search efficiency and showed promising results.

Chapter 8

General conclusion

8.1 Concluding remarks

The consideration of human factors in planning and scheduling of tasks in aircraft assembly lines may help to improve the management of both economic and ergonomic risks. For this purpose, the existing ergonomic methods have to be integrated in the mathematical models used in scheduling.

In this thesis, we considered in particular the case of Pulse Assembly Lines described in Chapter 1 with the objective to integrate the ergonomic factors in scheduling of assembly tasks. It was shown that this problem can be modelled as a special case of Project Resource-Constrained Scheduling Problem in terms of scheduling problem but it has never been considered under ergonomic constraints in the literature. First, in Chapter 2, we analyzed the ergonomic methods existing for the evaluation of physical workload and those that have been already used in scheduling applications in production systems. In particular, the physiological aspects of ergonomic risks were considered in detail. In Chapter 3, we presented the existing algorithms for lower bound calculation for RCPSP as well as some solution techniques.

In Chapter 4, a novel pseudo-polynomial algorithm to calculate a lower bound on makespan for RCPSP was presented. This approach is based on the evaluation of the highest possible usage of one resource usage subject to another resource. We also showed how this algorithm can be adapted for generalized RCPSP statement with piecewise-constant capacity function and continuous time. It was shown that for this algorithm the set of breakpoints can be used instead of time slots. This allows decreasing the complexity of the algorithm to polynomial $O(r^2n^2(n+m))$ number of operations, where n – number of tasks, r – number of resources, m – number of breakpoints in resource capacity function.

In Chapter 5, we proposed the techniques of propagation of resource capacity function and extensions of existing resource-based propagators (i.e. Edge-Finding, Extended Edge-Finding, Time Tabling) to generalized RCPSP statement with piecewise-constant capacity function. Some strengthening techniques were suggested for these propagators. It was shown that bounding of resource capacity can increase the propagation efficiency. It should be noted that all these new propagators can be used in generalized statement together with classic propagators based on precedence or disjunctive relations to make task domains tighter.

In Chapter 6, the first models integrating ergonomic factors in RCPSP have been proposed. Two different problem formulations were developed as generalized RCPSP with human factors constraints integrating ergonomic evaluation methods reviewed in Chapter 2. According to the conclusions based on the muscular activity discussed in Chapter 2, physical ergonomic constraints have been defined not only for planning horizon (takt time), but also for shorter intervals. The makespan objective function was considered for the first model and the cognitive load minimization as the objective function for the second model oriented to the reduction of the probability of operator error. Using these models for planning aircraft assembly process allows decreasing the negative impact of assembly process on workers' health. Numerical experiments on real data have shown the consistency of models.

Workforce planning and scheduling was considered in Chapter 7. The presented model takes into account assembly process constraints (including human factors) and helps to define the operator profiles to be recruited and trained. Proposed constraint programming model was tested with and without the use of constraint propagators developed in Chapter 5 in data preprocessing. The results of numerical experiments confirm that the model can be used to solve the optimization problem efficiently and the use of constraint propagators may help to reduce the solution time.

8.2 Future perspectives

The objective of this research was to develop a new scheduling approach capable of improving working conditions in aircraft final assembly lines due to an intelligent task allocation. The developed planning approach was tested on an industrial case studies and showed promising results in terms of solution time and objectives attained. All suggested models can be used as a base for future development of scheduling tools for aircraft assembly lines. A lot has to be done, since currently too few optimization problems are solved efficiently in practice, the decision making is mostly a manual process. In this study, an attempt was made to model a large variety of constraints and objective functions related to the aircraft assembly process including those related to human factors. However, for each particular assembly line configuration, some additional constraints and objectives may be relevant and properly modelled.

The further research can be in particular dedicated to the creation of new mathematical models and solution methods to handle unplanned disruptions (worker absence, late part supply or a quality problem) where rescheduling solution has to be found in a very short time in order to not compromise the respect of the tact time. Since the developed approach aims in respecting the regulatory ergonomic constraints and individual restrictions, but does not compare if all workers are equally charged in terms of working time or ergonomic burden, the objective of fairness of work distribution should be further formulated and studied.

Other research perspectives concern the development of constraint propagators, in particular, new polynomial-time propagators. It is interesting to go deeper into the idea of propagation of not only solution variables, but all input variables, which are used in constraints and objectives. Moreover, additional variables (i.e. disjunctions, resource usage upper bounds) can be used and propagated to enhance scheduling process efficiency. The synergistic effect of propagators is also to be explored.

Notations

Main notations used in this thesis is recalled in the following table.

N	set of tasks
R	set of renewable resources
p_j	processing time of task j
a_{jX}	required amount of resource $X \in R$ during the processing of task j
r_j	release time, the earliest time from which task j can be started
D_j	deadline, the latest time for completing task j
$c_r(t)$	piecewise constant capacity function of resource $r \in R$
est_j	the earliest starting time of task $j \in N$
ect_j	the earliest completion time of task $j \in N$
lst_j	the latest starting time of task $j \in N$
lct_j	the latest completion time of task $j \in N$
h_j	tail of job $j \in N$
$[lst_j, ect_j) \neq \emptyset$	compulsory part of task $j \in N$
N_r	set of tasks $N_r \subseteq N$ which require resource $r \in R$, i.e. $N_r = \{j \in N \| a_{jr} > 0\}$
e_{ij}	time lag between processing of tasks $i \in N$ and $j \in N$
$N^{CP} \subseteq N$	set of tasks with non-empty compulsory parts
$[est_j, lct_j)$	task processing domain
g_{ij}	disjunctive relation of pair of tasks $(i, j) \in N^2$
$g_{ij}(t)$	disjunctive function defined for pair of tasks $(i, j) \in N^2$
O	set of operators
S	set of specialties
b_{js}	number of operators with specialty $s \in S$ required to process task $j \in N$
M	set of methods to measure an ergonomic impact
erg_{mjs}	$(m \in M, j \in N, s \in S)$ – the ergonomic impact of task j on operator with specialty s evaluated by method m
U_{mo} (U^h_{mo})	the critical level of the total ergonomic impact evaluated by method m for all tasks processed by the same operator o
G	set of groups of tasks
$g_j \in G$	group which contains task $j \in N$

Chapter 9

Bibliography

[1] E. M. Frazzon, M. Kuck, and M. Freitag. Data-driven production control for complex and dynamic manufacturing systems. *CIRP Annals*, 67(1):515–518, 2018.

[2] G. Schuh, C. Reuter, J.P. Prote, F. Brambring, and J. Ays. Increasing data integrity for improving decision making in production planning and control. *CIRP Annals*, 66(1):425–428, 2017.

[3] Emmett J. Lodree, Christopher D. Geiger, and Xiaochun Jiang. Taxonomy for integrating scheduling theory and human factors: Review and research opportunities. *International Journal of Industrial Ergonomics*, 39(1):39–51, jan 2009.

[4] F. Mannhardt, S. A. Petersen, and M .F. Oliveira. Privacy challenges for process mining in human-centered industrial environments. In *2018 14th International Conference on Intelligent Environments (IE)*. IEEE, jun 2018.

[5] J.Takala, P. Hämäläinen, K. L. Saarela, L. Y. Yun, K. Manickam, T. W. Jin, P. Heng, C. Tjong, L. G. Kheng, S. Lim, and G. S. Lin. Global estimates of the burden of injury and illness at work in 2012. *Journal of Occupational and Environmental Hygiene*, 11(5):326–337, apr 2014.

[6] P. Hamalainen, J. Takala, and T. B. Kiat. Global estimates of occupational accidents and work-related illnesses 2017. Technical report, Workplace Safety and Health Institute, 2017.

[7] EU-OSHA. Estimating the costs of work -related accidents and ill-health: An analysis of european data sources. Technical report, Publications Office of the European Union, Luxembourg, 2017.

[8] E. H. Grosse, M. Calzavara, C. H. Glock, and F. Sgarbossa. Incorporating human factors into decision support models for production and logistics: current state of research. *IFAC-PapersOnLine*, 50(1):6900–6905, jul 2017.

[9] J. Raju. A conceptual design and cost optimisation methodology. In *44th AIAA/ASME/AHS Structure, Structural Dynamics and Materials Conference*. American Institute of Aeronautics and Astronautics, 2003.

[10] F. Mas, J. Rios, J. L. Menendez, and A. Gomez. A process-oriented approach to modeling the conceptual design of aircraft assembly lines. *The International Journal of Advanced Manufacturing Technology*, 67(1-4):771–784, sep 2012.

[11] J. Rios, F. Mas, and J. L. Menendez. Aircraft final assembly line balancing and workload smoothing: A methodological analysis. *Key Engineering Materials*, 502:19–24, 2012.

[12] Aircraftmakers. Keep seatbelts fastened. *The Economist*, June 20th 2015.

[13] AIRBUS. Airbus global market forecast 2017–2036—growing horizons. Technical report, AIRBUSS. A.S. , France, 2017.

[14] M. Urgo, J. Buergin, T. Tolio, and G. Lanza. Order allocation and sequencing with variable degree of uncertainty in aircraft manufacturing. *CIRP Annals*, 67(1):431–436, 2018.

[15] B. Carnahan, B. Norman, and M. Redfern. Incorporating physical demand criteria into assembly line balancing. *IIE Transactions*, 33(10):875–887, 2001.

[16] A. Otto and A. Scholl. Incorporating ergonomic risks into assembly line balancing. *European Journal of Operational Research*, 212(2):277–286, jul 2011.

[17] D. Battini, X. Delorme, A. Dolgui, A. Persona, and F. Sgarbossa. Ergonomics in assembly line balancing based on energy expenditure: a multi-objective model. *International Journal of Production Research*, 54(3):824–845, aug 2015.

[18] A. Otto and O. Battaia. Reducing physical ergonomic risks at assembly lines by line balancing and job rotation: a survey. *Computers and Industrial Engineering*, 111:467–480, 2017.

[19] M. Garey and D. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, 4(4):397—-411, 1975.

[20] OSHA. Osha ergonomics: The study of work. Technical report, U.S. Department of Labor, Occupational Safety and Health Administration,, 2000.

[21] A. E. Nixon, J. J. Mazzola, J. Bauer, J. R. Krueger, and P. E. Spector. Can work make you sick? a meta-analysis of the relationships between job stressors and physical symptoms. *Work & Stress*, 25(1):1–22, jan 2011.

[22] T. R. Waters, V. Putz-Anderson, A. Garg, and L. J. Fine. Revised niosh equation for the design and evaluation of manual lifting tasks. *Ergonomics*, 36:749–776, 1993.

[23] D. Liles and S. Deivanayagam. A job severity index for the evaluation an control of lifting injury. *Human Factors*, 26:683–693, 1984.

[24] L. McAtamney and E. N. Corlett. Rula: a survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, 24:19, 1993.

[25] S. Hignett and L. McAtamney. Rapid entire body assessment (reba). *Applied Ergonomics*, 31:201–205, 2000.

[26] E. Occhipinti. Ocra: A concise index for the assessment of exposure to repetitive movements of the upper limbs. *Ergonomics*, 41:1290–1311, 1998.

[27] J. S. Moore and A. Garg. The strain index: A proposed method to analyze jobs for risk of distal upper extremity disorders. *American Industrial Hygiene Association Journal*, 56:443–458, 1995.

[28] NIOSH. Occupational noise exposure: Revised criteria 1998. Technical report, U.S. Department of Health and Human Services, 1998.

[29] K. Schaub, B. Caragnano, G. Britzke, and R. Bruder. The european assembly worksheet. *Theoretical Issues in Ergonomics Science*, 14:616–639, 2013.

[30] A. Garg, D. B. Chan, and G. D. Herrin. Prediction of metabolic rates for manual materials handling jobs. *The American Industrial Hygiene Association Journal*, 39:661–674, 1978.

[31] R. Kolish and R. Padman. *An Integrated Survey of Project Scheduling.* 1997.

[32] Jan Weglarz, editor. *Project Scheduling.* Springer US, 1999.

[33] R. Kolisch and S. Hartmann. Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis. In *Project Scheduling*, pages 147–178. Springer US, 1999.

[34] S. Hartmann and R. Kolisch. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 127(2):394–407, dec 2000.

[35] Frederick S. Hillier, editor. *Project Scheduling.* Kluwer Academic Publishers, 2002.

[36] Christoph Schwindt and Jürgen Zimmermann, editors. *Handbook on Project Management and Scheduling Vol.1.* Springer International Publishing, 2015.

[37] O. Kone, C. Artigues, P. Lopez, and M. Mongeau. Event-based MILP models for resource-constrained project scheduling problems. *Computers & Operations Research*, 38(1):3–13, jan 2011.

[38] C. Artigues, O. Kone, P.Lopez, and M. Mongeau. Mixed-integer linear programming formulations. In *Handbook on Project Management and Scheduling Vol.1*, pages 17–41. Springer International Publishing, oct 2014.

[39] R. Kolisch and A. Sprecher. PSPLIB - a project scheduling problem library: OR software - ORSEP operations research software exchange program. *European Journal of Operational Research*, 96(1):205–216, 1997. http://www.om-db.wi.tum.de/psplib/.

[40] Neron E., Christian Artigues, Philippe Baptiste, Jacques Carlier, Jean Damay, Sophie Demassey, and Philippe Laborie. Lower bounds for resource constrained project scheduling problem. In Joanna Jozefowska and Jan Weglarz, editors, *Perspectives in modern project scheduling*, chapter 7, pages 167–204. Springer, 2006.

[41] S. Knust. Lower bounds on the minimum project duration. In Christoph Schwindt and Jurgen Zimmermann, editors, *Handbook on Project Management and Scheduling*, volume 1, chapter 3, pages 43–56. Springer, 2015.

[42] E. Gafarov, A. Lazarev, and F. Werner. On lower and upper bounds for the resource-constrained project scheduling problem. Technical report, Otto-von-Guericke Universitaet, 2010. 27 p.

[43] P. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-Based Scheduling*. Kluwer Academic Publisher, 2001.

[44] P. Laborie. Algorithms for propagation of resource constraints in AI planning and scheduling: Existing approaches and new results. *Artificial Intelligence*, 143:151–188, 2003.

[45] Vilim P. *Global Constraints in Scheduling*. PhD thesis, Charles University in Prague, Faculty of Mathematics, Physics, Department of Theoretical Computer Science, and Mathematical Logic, 2007.

[46] E. A. Attia, V. Dumbrava, and P. Duquenne. Factors affecting the development of workforce versatility. *IFAC Proceedings Volumes*, 45(6):1221–1226, may 2012.

[47] S. Anil Kumar and N. Suresh. *Operations Management*. New Age International, 2009.

[48] M. Baudin. *Lean Assembly – the Nuts and Bolts of Making Assembly Operation Flow*. Productivity Press, 2002.

[49] J. Siegrist and N. Dragano. Pension at 67 - problems and challenges from a health science perspective. *Working Paper 147*, 2007.

[50] R. E. Gunther, G. D. Johnson, and R. S. Peterson. Currently practiced formulations for the assembly line balance problem. *Journal of Operations Management*, 3(4):209–221, aug 1983.

[51] D. Woods, D. Fisher, and R. Andres. Minimizing fatigue during repetitive jobs: Optimal work rest schedules. *Human Factors*, 39:83–101, 1997.

[52] C. Jaturanonda and S. Nanthavanij. Heuristic procedure for two-criterion assembly line balancing problem. *Industrial Engineering & Management Systems*, 5:84–96, 2006.

[53] G. Choi. A goal programming mixed-model line balancing for processing time and physical workload. *Computers & Industrial Engineering*, 57(1):395–400, aug 2009.

[54] H. Rajabalipour Cheshmehgaz, H. Haron, F. Kazemipour, and M. Ishak Desa. Accumulated risk of body postures in assembly line balancing problem and modeling through a multi-criteria fuzzy-genetic algorithm. *Computers and Industrial Engineering*, 63(2):503–512, sep 2012.

[55] O. Mutlu and E. Ozgormus. A fuzzy assembly line balancing problem with physical workload constraints. *International Journal of Production Research*, 50(18):5281–5291, sep 2012.

[56] Z. Xu, J. Ko, D. J. Cochran, and M.-C. Jung. Design of assembly lines with the concurrent consideration of productivity and upper extremity musculoskeletal disorders using linear models. *Computers andIndustrial Engineering*, 62(2):431–441, mar 2012.

[57] ACGIH. *TLVs and BEIs guidelines.* American Conference of Governmental Industrial Hygienists, 2010.

[58] C. Jaturanonda, S. Nanthavanij, and S. K. Das. Heuristic procedure for the assembly line balancing problem with postural load smoothness. *International Journal of Occupational Safety and Ergonomics*, 19(4):531–541, jan 2013.

[59] Y. Kara, Y. Atasagun, H. Gokcen, S. Hezer, and N. Demirel. An integrated model to incorporate ergonomics and resource restrictions into assembly line balancing. *International Journal of Computer Integrated Manufacturing*, 27(11):997–1007, jan 2014.

[60] J. Sternatz. Enhanced multi-hoffmann heuristic for efficiently solving real-world assembly line balancing problems in automotive industry. *European Journal of Operational Research*, 235(3):740–754, jun 2014.

[61] N. Barathwaj, P. Raja, and S. Gokulraj. Optimization of assembly line balancing using genetic algorithm. *Journal of Central South University*, 22(10):3957–3969, oct 2015.

[62] S. D. Akyol and A. Baykasoglu. ErgoALWABP: a multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem under ergonomic risk factors. *Journal of Intelligent Manufacturing*, 30(1):291–302, jul 2016.

[63] J. Bautista, R. Alfaro-Pozo, and C.Batalla-Garcia. Maximizing comfort in assembly lines with temporal, spatial and ergonomic attributes. *International Journal of Computational Intelligence Systems*, 9(4):788–799, jun 2016.

[64] J. Bautista, C. Batalla-Garcia, and R. Alfaro-Pozo. Models for assembly line balancing by temporal, spatial and ergonomic risk attributes. *European Journal of Operational Research*, 251(3):814–829, jun 2016.

[65] B. J. Carnahan, M. S. Redfern, and B. Norman. Designing safe job rotation schedules using optimization and heuristic search. *Ergonomics*, 43(4):543–560, apr 2000.

[66] S. Nanthavanij and T. Kullpattaranirun. A genetic algorithm approach to determine minimax work assignments. *International Journal of Industrial Engineering - Theory Applications and Practice*, 8:176–185, 2001.

[67] W. Tharmmaphornphilas, B. Green, B. J. Carnahan, and B. A. Norman. Applying mathematical modeling to create job rotation schedules for minimizing occupational noise exposure. *AIHA Journal*, 64(3):401–405, may 2003.

[68] K. Yaoyuenyong and S. Nanthavanij. A modified lpt swap heuristic for solving large minimax work assignment problems. *Industrial Engineering and Management Systems*, 2:121–130, 2003.

[69] W. Tharmmaphornphilas and B. A. Norman. A quantitative method for determining proper job rotation intervals. *Annals of Operations Research*, 128(1-4):251–266, apr 2004.

[70] T. Kullpattaranirun and S. Nanthavanij. A heuristic genetic algorithm for solving complex safety-based work assignment problems. *International Journal of Industrial Engineering – Theory Applications and Practice*, 12:45–57, 2005.

[71] K. Asawarungsaengkul and S. Nanthavanij. Design of optimal noise hazard control strategy with budget constraint. *International Journal of Occupational Safety and Ergonomics*, 12(4):355–367, jan 2006.

[72] S. Yaoyuenyong and S. Nanthavanij. Hybrid procedure to determine optimal workforce without noise hazard exposure. *Computers and Industrial Engineering*, 51:743–764, 2006.

[73] W. Tharmmaphornphilas and B. A. Norman. A methodology to create robust job rotation schedules. *Annals of Operations Research*, 155(1):339–360, jul 2007.

[74] S. U. Seckiner and M. Kurt. A simulated annealing approach to the solution of job rotation scheduling problems. *Applied Mathematics and Computation*, 188(1):31–45, may 2007.

[75] K. Asawarungsaengkul and S. Nanthavanij. Heuristic genetic algorithm for workforce scheduling with minimum total worker-location changeover. *International Journal of Industrial Engineering*, 15:373–385, 2008.

[76] K. Asawarungsaengkul and S. Nanthavanij. Optimization approach to hazard prevention budgeting problem. *International Journal of Industrial Engineering*, 15:330–340, 2008.

[77] S. U. Seckiner and M. Kurt. Ant colony optimization for the job rotation scheduling problem. *Applied Mathematics and Computation*, 201(1-2):149–160, jul 2008.

[78] S. Yaoyuenyong and S. Nanthavanij. Heuristic job rotation procedures for reducing daily exposure to occupational hazards. *International Journal of Occupational Safety and Ergonomics*, 14(2):195–206, jan 2008.

[79] M. B. Aryanezhad, A. S. Kheirkhah, V. Deljoo, and S. M. J. Mirzapour Al e hashem. Designing safe job rotation schedules based upon workers' skills. *The International Journal of Advanced Manufacturing Technology*, 41(1-2):193–199, mar 2008.

[80] J.A. Diego-Mas, S. Asensio-Cuesta, M.A. Sanchez-Romero, and M.A. Artacho-Ramirez. A multi-criteria genetic algorithm for the generation of job rotation schedules. *International Journal of Industrial Ergonomics*, 39(1):23–33, jan 2009.

[81] G. Michalos, S. Makris, L. Rentzos, and G. Chryssolouris. Dynamic job rotation for workload balancing in human based assembly systems. *CIRP Journal of Manufacturing Science and Technology*, 2(3):153–160, jan 2010.

[82] G. Michalos, S. Makris, and D. Mourtzis. A web based tool for dynamic job rotation scheduling using multiple criteria. *CIRP Annals*, 60(1):453–456, 2011.

[83] S. Nanthavanij, S. Yaoyuenyong, and C. Jeenanunta. Heuristic approach to workforce scheduling with combined safety and productivity objective. *International Journal of Industrial Engineering*, 17(4):319–333, 2010.

[84] S. Asensio-Cuesta, J. A. Diego-Mas, L. Canos-Daros, and C. Andres-Romano. A genetic algorithm for the design of job rotation schedules considering ergonomic and competence criteria. *The International Journal of Advanced Manufacturing Technology*, 60(9-12):1161–1174, oct 2011.

[85] S. H. Rodgers. A functional job analysis technique. *Occupational medicine (Philadelphia, Pa.)*, 7:679–711, 1992.

[86] S. Asensio-Cuesta, J.A. Diego-Mas, L.V. Cremades-Oliver, and M.C. Gonzalez-Cruz. A method to design job rotation schedules to prevent work-related musculoskeletal disorders in repetitive work. *International Journal of Production Research*, 50(24):7467–7478, dec 2012.

[87] G. Michalos, S. Makris, and G.Chryssolouris. The effect of job rotation during assembly on the quality of final product. *CIRP Journal of Manufacturing Science and Technology*, 6(3):187–197, jan 2013.

[88] L. Ma, D. Chablat, F. Bennis, and W. Zhang. A new simple dynamic muscle fatigue model and its validation. *International Journal of Industrial Ergonomics*, 39(1):211–220, jan 2009.

[89] A. Otto and A. Scholl. Reducing ergonomic risks by job rotation scheduling. *OR Spectrum*, 35(3):711–733, apr 2012.

[90] G. Mossa, F. Boenzi, S. Digiesi, G. Mummolo, and V.A. Romano. Productivity and ergonomic risk in human based production systems: A job-rotation scheduling model. *International Journal of Production Economics*, 171:471–477, jan 2016.

[91] S.-Y. Yoon, J. Ko, and M .C. Jung. A model for developing job rotation schedules that eliminate sequential high workloads and minimize between-worker variability in cumulative daily workloads: Application to automotive assembly lines. *Applied Ergonomics*, 55:8–15, jul 2016.

[92] J. Song, C. Lee, W. Lee, S. Bahn, C. Jung, and M. H. Yun. Development of a job rotation scheduling algorithm for minimizing accumulated work load per body parts. *Work*, 53:511–521, 2016.

[93] A. Enomoto, N. Yamamoto, and T. Suzuki. Automatic estimation of the ergonomics parameters of assembly operations. *CIRP Annals*, 62(1):13–16, 2013.

[94] J. Kruger and T. D. Nguyen. Automated vision-based live ergonomics analysis in assembly operations. *CIRP Annals*, 64(1):9–12, 2015.

[95] C.H. Glock, E.H. Grosse, T. Kim, W.P. Neumann, and A. Sobhani. An integrated cost and worker fatigue evaluation model of a packaging process. *International Journal of Production Economics*, 207:107–124, jan 2019.

[96] V. Visentin, F. Sgarbossa, M. Calzavara, and A. Persona. Fatigue accumulation in the assignment of manual material handling activities to operators. *IFAC-PapersOnLine*, 51(11):826–831, 2018.

[97] M. Baucells and L. Zhao. It is time to get some rest. *Management Science*, 65(4):1717–1734, apr 2019.

[98] A. F. Huxley. Muscle structure and theories of contraction. *Progress in Biophysics and Biophysical Chemistry*, 7:255–318, 1957.

[99] E. I. Chazov, V. N. Smirnov, V. A. Saks, and L. V. Rozenshtrauh. The phenomenon of regulation of the force of contraction of the heart muscle by creatine. Technical report, Cardiological scientific center of the Academy of Medical Sciences of the USSR, 1973.

[100] A. Lahrichi. Ordonnancements. La notion de "parties obligatoires" et son application aux problemes cumulatifs. *RAIRO-Operations Research*, 16:241–262, 1982.

[101] B. Roy and M. A. Sussman. Les problemes d'ordonnancement avec contraintes disjonctive. Tech. rep. Note DS 9, SEMA, Paris, 1964.

[102] A. Schutt, T. Feydy, P. Stuckey, and M. Wallace. Explaining the cumulative propagator. *Constraints*, 16(3):250–282, 2011.

[103] P. Baptiste and C. Le Pape. Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems. *Constraints*, 5:119–139, 2000.

[104] J. Carlier and E. Pinson. An algorithm for solving the job-shop problem. *Management Science*, 35:164–176, 1989.

[105] J. Carlier and E. Pinson. A practical use of jackson's preemptive schedule for solving the job-shop problem. *Annals of Operations Research*, 26:269–287, 1990.

[106] D. Applegate and W. Cook. A computational study of job-shop scheduling. *ORSA Journal on Computing*, 3(2):149–156, 1991.

[107] Ph. Baptiste, C. Le Pape, and W. Nuijten. Satisfiability tests and timebound adjustments for cumulative scheduling problems. *Annals of Operations Research*, 92:305–333, 1999.

[108] J. Carlier and E. Pinson. Adjustment of heads and tails for the job-shop problem. *European Journal of Operational Research*, 78:146–161, 1994.

[109] J. Carlier and B Latapie. Une methode arborescente pour resoudre les problemes cumulatifs. *RAIRO-Recherche Operationnelle*, 25:311–340, 1991.

[110] J. Carlier and E. Pinson. Jackson's pseudo preemptive schedule for the $pm/r_i, q_i/c_{max}$ scheduling problem. *Annals of Operations Research*, 83:41–48, 1998.

[111] J. Carlier and E. Pinson. Jackson's pseudo preemptive schedule and cumulative scheduling problems. *Discrete Applied Mathematics*, 145:80–94, 2004.

[112] P Brucker. *Scheduling algorithms.* Springer Verlag, Berlin, 2002.

[113] M. Haouari and A. Gharbi. An improved max-flow-based lower bound for minimizing maximum lateness on identical parallel machines. *Operations Research Letters*, 31:49–52, 2003.

[114] P. Tercinet, Ch. Lente, and E. Neron. Mixed satisfiability tests for multiprocessor scheduling with release dates and deadlines. *Operations Research Letters*, 32(11):326–330, 2004.

[115] A. Mingozzi, V. Maniezzo, S. Ricciardelli, and L. Bianco. An exact algorithm for the multiple resource-constrained project scheduling problem based on a new mathematical formulation. *Management Science*, 44:714–729, 1998.

[116] P. Brucker and S. Knust. A linear programming and constraint propagation based lower bound for the RCPSP. *European Journal of Operational Research*, 127:355–362, 2000.

[117] J. Carlier and E. Neron. A new LP based lower bound for the cumulative scheduling problem. *European Journal of Operational Research*, 127(2):363–382, 2000.

[118] J. Carlier and E. Neron. On linear lower bounds for the resource constrained project scheduling problem. *European Journal of Operational Research*, 149:314–324, 2003.

[119] J. Erschler, P. Lopez, and C. Thuriot. Raisonnement temporel sous contraintes de ressources et problemes d'ordonnancement. *Revue d'Intelligence Artificielle*, 5:7–32, 1991.

[120] P. Lopez, J. Erschler, and P. Esquirol. Ordonnancement de taches sous contraintes: une approche energetique. *Automatique-productique informatique industrielle*, 26(5-6):453–481, 1992.

[121] Ch. Schwindt. *Resource Allocation in Project Management.* Springer Verlag, 2005.

[122] W. Nuijten. *Time and resource constrained scheduling: A constraint satisfaction approach.* PhD thesis, Eindhoven University of Technology, 1994.

[123] Y. Caseau and F. Laburthe. Cumulative scheduling with task intervals. In *Proceedings of the Joint International Conference and Symposium on Lower bounds for Resource Constrained Project Scheduling Problem Logic Programming*, pages 363–377. The MIT Press, Cambridge, 1996.

[124] T. Garaix, Ch. Artigues, and S. Demassey. Bornes inferieures et superieures pour le RCPSP. In *Proceedings of 6ieme congres de la Societe Francaise de Recherche Operationnelle et d'Aide a la Decision*, pages 219–240, Tours, France, 2005.

[125] P. Baptiste and S. Demassey. Tight LP bounds for resource constrained project scheduling. *OR Spectrum*, 26:251–262, 2004.

[126] P. Laborie. Complete MCS-based search: Application to resource constrained project scheduling. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2005.

[127] A. A. Pritsker, L. J. Watters, and P. M. Wolfe. Multi-project scheduling with limited resources: a zero-one programming approach. *Management Science*, 16:93–108, 1969.

[128] N. Christofides, R. Alvarez-Valdes, and J. M. Tamarit. Project scheduling with resource constraints: a branch and bound approach. *European Journal of Operational Research*, 29(3):262–273, 1987.

[129] C. Le Pape. *Des systemes d'ordonnancement flexibles et opportunistes.* PhD thesis, Universite Paris XI, 1988.

[130] B. R. Fox. Non-chronological scheduling. In *Proceedings of AI, Simulation and Planning in High Autonomy Systems*, pages 72–77, 1990.

[131] N. Beldiceanu and M. Carlsson. Sweep as a generic pruning technique applied to the non-overlapping rectangles constraint. In *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming*, pages 377–391, 2001.

[132] N. Beldiceanu and M. Carlsson. A new multi-resource cumulatives constraint with negative heights. In *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*, pages 63–79, 2002.

[133] A. Letort, N. Beldiceanu, and M. Carlsson. A scalable sweep algorithm for the cumulative constraint. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming*, pages 439–454, 2012.

[134] N. Beldiceanu, M. Carlsson, and E. Poder. New filtering for the cumulative constraint in the context of non-overlapping rectangles. In *Proceedings of the 5th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 21–35, 2008.

[135] S. Gay, R. Hartert, and P. Schaus. Simple and scalable time-table filtering for the cumulative constraint. In *Proceedings of the 21st International Principles and Practice of Constraint Programming CP*, pages 149–157, 2015.

[136] P. Vilim. Edge finding filtering algorithm for discrete cumulative resources in o(kn log n). In *Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming*, pages 802–816, 2009.

[137] R. Kameugne, L. Fotso, and J. Scott. A quadratic edge-finding filtering algorithm for cumulative resource constraints. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming CP*, pages 478–492, 2011.

[138] L. Mercier and P. Van Hentenryck. Edge finding for cumulative scheduling. *INFORMS Journal on Computing*, 20(1):143–153, 2008.

[139] P. Vilim. Timetable edge finding filtering algorithm for discrete cumulative resources. In *Proceedings of the 8th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 230–245, 2011.

[140] P. Ouellet and C. Quimper. Time-table extended-edge-finding for the cumulative constraint. In *Proceedings of the 19th International Conference on Principles and Practice of Constraint Programming CP*, pages 562–577, 2013.

[141] A. Schutt, T. Feydy, and P.J. Stuckey. Explaining time-table-edge-finding propagation for the cumulative resource constraint. In *Proceedings of the 10th International Conference on Integration of AI and OR Techniques and Constraint Programming for Combinatorial Optimization Problems*, 2013.

[142] P. Baptiste, C. Le Pape, and Nuijten W. *Constraint-Based Scheduling*. Kluwer Academic Publishers, 1998.

[143] A. Pritsker, B. Alan, and L. J. Watters. A zero-one programming approach to scheduling with limited resources. Santa Monica, CA: RAND Corporation, 1968. https://www.rand.org/pubs/research$_{m}emoranda/RM5561.html$.

[144] L. Bianco and M. Caramia. A new formulation for the project scheduling problem under limited resources. *Flexible Services and Manufacturing Journal*, 25(1-2):6–24, nov 2011.

[145] E. L. Lawler. On scheduling problems with deferral costs. *Management Science*, 11(2):280–288, nov 1964.

[146] L.A. Kaplan. *Resource-constrained project scheduling with preemption of jobs*. PhD thesis, University of Michigan, 1998. Unpublished PhD Dissertation.

[147] R. Klein. *Scheduling of Resource-Constrained Projects*. Springer US, 2000.

[148] A. Kimms. *Mathematical Programming and Financial Objectives for Scheduling Projects*. Springer US, 2001.

[149] J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34(3):391–401, mar 1988.

[150] F. Stork and M. Uetz. On the generation of circuits and minimal forbidden sets. *Mathematical Programming*, 102(1):205–205, may 2004.

[151] C. Artigues, P. Michelon, and S. Reusser. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149(2):249–267, sep 2003.

[152] M. Queyranne and A. S. Schulz. Polyhedral approaches to machine scheduling. Technical Report 408, Technische Universitat Berlin, 1994.

[153] J. C. Zapata, B. M. Hodge, and G. V. Reklaitis. The multimode resource constrained multiproject scheduling problem: Alternative formulations. *AIChE Journal*, 54(8):2101–2119, aug 2008.

[154] S. Mouret, I. E. Grossmann, and P. Pestiaux. Time representations and mathematical models for process scheduling problems. *Computers & Chemical Engineering*, 35(6):1038–1063, jun 2011.

[155] T. S. Kyriakidis, G. M. Kopanos, and M. C. Georgiadis. MILP formulations for single- and multi-mode resource-constrained project scheduling problems. *Computers & Chemical Engineering*, 36:369–385, jan 2012.

[156] C. Artigues, P. Brucker, S. Knust, O. Kone, P. Lopez, and M. Mongeau. A note on "event-based MILP models for resource-constrained project scheduling problems". *Computers & Operations Research*, 40(4):1060–1063, apr 2013.

[157] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, (1):269–271, 1959.

[158] B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*, 24(1):1–13, 1977.

[159] P. De Bruecker, J. Van den Bergh, J. Belien, and E. Demeulemeester. Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16, may 2015.