המחלקה למתמטיקה
Department of Mathematics

פרויקט מסכם לתואר בוגר במדעים (B.Sc) במתמטיקה שימושית

שיטת דאגלס-רצפורד ושיטת אלטרנציה של פון-נוימן לפתירת בעיית הקצאת משאבים

יקיר פרץ

The Douglas-Rachford and the Von Neumann Alternating Projections Methods for Solving the Unary Resource Constraint Problem

Yakir Peretz

פרויקט מסכם לתואר בוגר במדעים (B.Sc) במתמטיקה שימושית

שיטת דאגלס-רצפורד ושיטת אלטרנציה של פון-נוימן לפתירת בעיית הקצאת משאבים

יקיר פרץ

The Douglas-Rachford and the Von Neumann Alternating Projections Methods for Solving the Unary Resource Constraint Problem

Yakir Peretz

Advisor:
Dr. Aviv Gibali

מנחה:
ד"ר אביב גיבלי

Karmiel

כרמיאל

2019

המכללה האקדמית להנדסה אורט בראודה (ע"ר) רח' סנונית 51 ת.ד. 78 כרמיאל, 21982 טל' 04-9901911 פקס 04-9882016
Ort Braude College, P.O.Box 78, Karmiel 21982, Israel, Tel: 972-4-99901911, Fax: 972-4-9882016

Contents

List of Figures

Abstract

In this project we are concern with the *Convex Feasibility Problem* (CFP) which stands at the core of many real-world problem reformulations.

We are focus on a specific scheduling task which is known as the *Unary Resource Constraint problem* and propose a new reformulation of the problem as a feasibility problem. This reformulation allows to apply the class of projection methods and in particular we choose the Douglas-Rachford and Von Neumann Alternating Projections Algorithms which attracts much attention in recent years due to their effectiveness.

I would like to thank the mathematics department of ORT Braude College for being so helpful and supportive during my bachelor studies. A special gratitude is to Dr. Aviv Gibali who guide me through this interesting project. His time, effort and support along the way yielded the success of this work.

1 Introduction

In this project we are concern with the the Convex Feasibility Problem (CFP) which is phrased as follows. For $i = 0, 1, \cdots, m-1$, let $C_i \subseteq \mathbb{R}^n$ be nonempty, closed and convex sets. The CFP is formulated as follows.

$$\text{find a point } x^* \in C := \cap_{i=0}^{m-1} C_i. \tag{1.1}$$

The CFP has been used to model significant real-world problems in imaging, sensor networks, radiation therapy treatment planning, resolution enhancement and in many others; see e.g., [5]. One of the successful class of iterative methods for solving CFPs are known as Projection Methods. These are iterative algorithms that use projections onto sets, relying on the principle that when a family of sets is present, then projections onto the given individual sets are easier to perform than projections onto other sets (intersections, image sets under some transformation, etc.) that are derived from the given individual sets. Their main advantage, which makes them successful in real-world applications, is computational. They commonly are able to handle huge-size problems of dimensions beyond which more sophisticated methods cease to be efficient or even applicable due to memory requirements (see, e.g., [9, 11]). See illustrations of different types of projection methods in Figure 1 which is taken from [10].

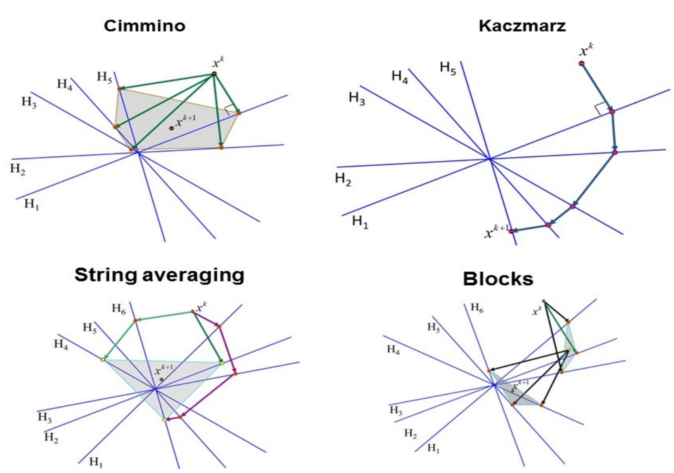


Figure 1: Different projection methods for the linear case. The figure is reproduced from [10]

Two specific algorithms of interest in the class of projection methods, are the *Douglas-Rachford* (DR) [12] and Von Neumann *Alternating Projections* method [16]. The Douglas-Rachford algorithm was originally proposed for solving a system of linear equations arising in heat conduction problems. Lions and Mercier [15] were the ones who made the major work in this field and adjusted and extended the algorithm successfully for solving CFPs and even more general problems, such as zero of the sum of two maximally monotone operators. For further and deeper investigation and generalization the readers are referred to the works of Bauschke et al. for example, [6, 2, 8, 7] and the references therein.

The Alternating Projections Algorithm is designed for best approximation problem, that is finding the projection onto the intersection of two closed subspaces in Hilbert space. The interested reader is referred to [4, 5, 9]. Observe that while both methods are designed for solving two-sets CFPs, there exist many techniques and generalizations which enable to apply the algorithms for solving the general CFP (1.1) with any finite number of sets. For example, two major examples which are explained in details later are the product space reformulation [17] and the Cyclic Douglas-Rachford Algorithm [3].

In this project we propose a new reformulation of the Unary Resource Constraint problem as a feasibility problem and then suggest how the Douglas-Rachford and the Alternating Projections algorithms can be applied for solving it.

2 Preliminaries

We start with several definitions and notions.

Definition 2.1 *Let $C \subset \mathbb{R}^n$ be a closed and convex set.*

*(i) The **`closest point projection`** in C is a mapping $P_C : \mathbb{R}^n \to C$ which assign for any $x \in \mathbb{R}^n$ an element denoted by $P_C(x)$ and is characterized by the fact that $P_C(x) \in C$ and is the solution of the following optimization problem*

$$P_C(x) = \mathrm{Argmin}\{\|z - x\| \mid z \in C\}. \tag{2.1}$$

It is known that for any point $x \in \mathbb{R}^n$, the following characterization of the projection hold.
i) $\|P_C(x) - P_C(y)\|^2 \leq \langle P_C(x) - P_C(y), x - y\rangle \ \forall y \in C$;
ii) $\|P_C(x) - y\|^2 \leq \|x - y\|^2 - \|x - P_C(x)\|^2 \ \forall y \in C$;
iii) $\langle (I - P_C)x - (I - P_C)y, x - y\rangle \geq \|(I - P_C)x - (I - P_C)y\|^2 \ \forall y \in C$.
For properties of the metric projection, the interested reader could be referred to [14, Section 3].

*(ii) The **`reflection`** in the set C is a mapping $R_C : \mathbb{R}^n \to \mathbb{R}^n$ defined as $R_C(x) = 2P_C(x) - x$.*

Several important examples in which the projection onto the sets has a close formula is given next.

Example 2.2 *(1) Projection onto half-space. Let the half-space*

$$H := \{z \in \mathbb{R}^n \mid \langle z, a\rangle \leq \beta\} \tag{2.2}$$

where $a \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$. The projection onto H is given as following.

$$P_H(x) = \begin{cases} x - \frac{\langle x,a\rangle - \beta}{\|a\|^2} a & \text{if } \langle x, a\rangle > \beta \\ x & \text{if } \langle x, a\rangle \leq \beta. \end{cases} \tag{2.3}$$

(2) Projection onto a box. A box in $\mathbb{R}^n$ is a Cartesian product of closed intervals,

$$\Box := \prod_{i=1}^{n} [\alpha_i, \beta_i] \tag{2.4}$$

where α_i and $\beta_i \in \mathbb{R}$. The projection onto $\Box$ is given as following.

$$[P_{\Box}(x)]_i = \begin{cases} x_i & \text{if } x_i \in [\alpha_i, \beta_i] \\ \beta_i & x_i > \beta_i \\ \alpha_i & x_i < \alpha_i. \end{cases} \tag{2.5}$$

In particular the projection onto the non-negative orthant $\mathbb{R}^n_+$ is obtained by choosing $\alpha_i = 0$ and $\beta_i = +\infty$ for all $i = 1, \ldots, n$.

$$\left[P_{\mathbb{R}^n_+}(x)\right]_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{else.} \end{cases} \tag{2.6}$$

Proofs for the above formulas can be found for example in [9, 4.1.3] and for the convenient of the reader as well as it usage to our analysis we bring next proof of the first example.

Proof. It is clear in (2.3) that if $\langle x, a \rangle \leq \beta$ then $P_H(x) = x$. On the other hand, if $\langle x, a \rangle < \beta$, denote by $y := x - (\langle x, a \rangle - \beta)/(\|a\|^2)a$. Clearly $\langle y, a \rangle = \beta$ so $y \in H$. Now for some $y \in H$ we have

$$\langle x - y, z - y \rangle = \frac{\langle x, a \rangle - \beta}{\|a\|^2} \left(\langle z, a \rangle - \langle y, a \rangle\right) \leq 0. \tag{2.7}$$

Using the characterization of the metric projection and the fact that $P_H(y) = y$ we get that

$$P_H(x) = x - \frac{(\langle x, a \rangle - \beta)_+}{\|a\|^2} a \tag{2.8}$$

where $(w)_+ := \max\{w, 0\}$ and this completes the proof. ■

Definition 2.3 *Let $T : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be an operator and let $C \subset \mathbb{R}^n$.*

*(i) The **fixed point set** of T denoted by $\mathrm{Fix}(T)$ is defined as*

$$\mathrm{Fix}(T) := \{x \in \mathbb{R}^n \mid T(x) = x\}. \tag{2.9}$$

*(ii) The operator T is called **nonexpansive** if*

$$\|T(x) - T(y)\| \leq \|x - y\| \text{ for all } x, y \in \mathbb{R}^n. \tag{2.10}$$

*(iii) The operator T is called **firmly nonexpansive** on C if*

$$\langle T(x) - T(y), x - y \rangle \geq \|T(x) - T(y)\|^2 \text{ for all } x, y \in C. \tag{2.11}$$

The next remarks are essential for our convergence analysis and it is quite straight forward which relies on the above definitions.

Remark 2.4 *(i) In case that $C \subset \mathbb{R}^n$ is convex, then both P_C and R_C are single-valued operators and moreover P_C is firmly nonexpansive and R_C is nonexpansive.*

(ii) Let $C_1, C_2 \subset \mathbb{R}^n$ be non-empty, closed and convex. Define the operator

$$T_{C_1,C_2} := 1/2\left(R_{C_2}R_{C_1} + I\right) = P_{C_2}\left(2P_{C_1} - I\right) + \left(I - P_{C_2}\right) \tag{2.12}$$

then T_{C_1,C_2} is firmly nonexpansive and

$$\operatorname{Fix}\left(T_{C_1,C_2}\right) = \left\{\bar{x} \in \mathbb{R}^n \mid P_{C_2}(\bar{x}) \in C_1 \cap C_2\right\}. \tag{2.13}$$

The operator T_{C_1,C_2} is known in the literature as the ***Douglas-Rachford operator****.*

Following the above we can define the Douglas-Rachford Algorithm. Let $A, B \subseteq \mathbb{R}^n$ be non-empty, closed and convex sets.

Algorithm 1 The Douglas-Rachford Algorithm for 2-sets CFP

Initialization: Choose an arbitrary initial point $x_0 \in \mathbb{R}^n$ and set $k = 0$.
Iteration step: Given the current iterate x_k, calculate the next iterate as

$$x_{k+1} = T_{A,B}(x_k) \tag{2.14}$$

Another related algorithm which is used in our numerical experiments as comparison is the Von Neumann Alternating Projections Algorithm [16]. The method is designed for best approximation problem, that is finding the projection onto the intersection of two closed subspaces in Hilbert space. Let $\mathcal{H}$ be a real Hilbert space, and let A and B be closed subspaces. Choose $x \in \mathcal{H}$ and construct the sequences $\{a_k\}_{k=0}^{\infty}$ and $\{b_k\}_{k=0}^{\infty}$ by

$$\begin{cases} b_0 = x, \\ a_k = P_A(b_{k-1}) \text{ and } b_k = P_B(a_k), \ k = 1, 2, \ldots, \end{cases} \tag{2.15}$$

where P_A and P_B denote the orthogonal projection operators of $\mathcal{H}$ onto A and B, respectively. Von Neumann showed [16, Lemma 22] that both sequences

$\{a_k\}_{k=0}^{\infty}$ and $\{b_k\}_{k=0}^{\infty}$ converge strongly to $P_{A\cap B}(x)$. This algorithm is known as von Neumann's alternating projections method. Observe that not only the sequences converge strongly, but also that their common limit is the nearest point to x in $A \cap B$.

See geometrical interpolation of the Douglas-Rachford and the Alternating Projections iterative steps are presented in Figures 2 and 3 (taken Dr. D. Rubén Campoy García thesis [13]).

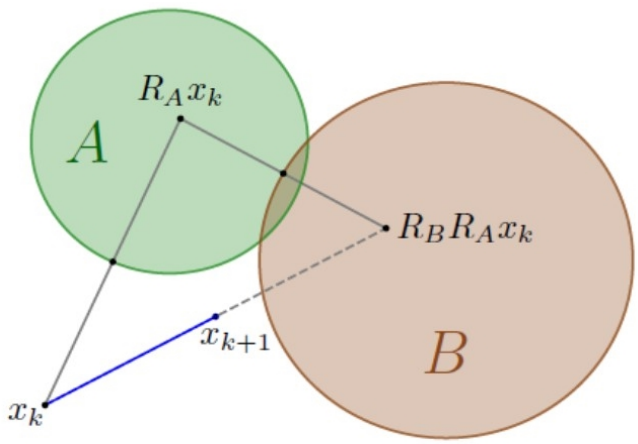


Figure 2: The iterative step of the Douglas-Rachford algorithm with the sets A and B.

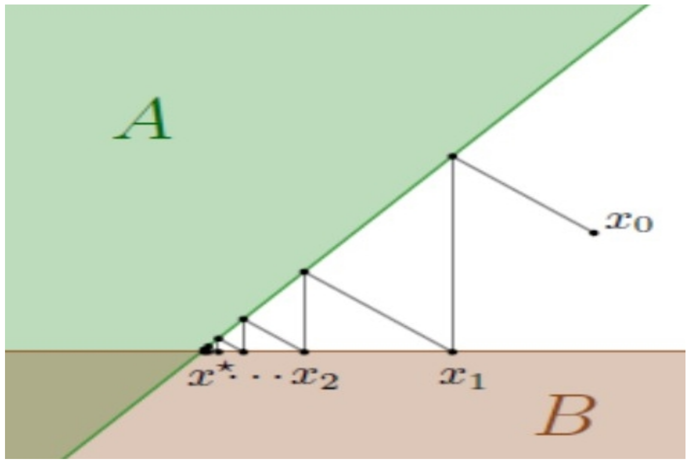


Figure 3: The Alternating Projections algorithm with the subspaces A and B.

The convergence of the Douglas-Rachford and the Alternating Projections Algorithms follows directly from the below *Opial's Theorem* which is also known in the literature as the *Krasnosel'skiĭ-Mann Theorem*.

Theorem 2.5 *Let $\mathcal{H}$ be a real Hilbert space and $C \subset \mathcal{H}$ be closed and convex. Assume that $T : C \to C$ is firmly nonexpansive with $\mathrm{Fix}(T) \neq \emptyset$. Then, for an arbitrary $x_0 \in C$, the sequence $\{x_{k+1} = T(x_k)\}_{k=0}^{\infty}$ converges weakly to $z \in \mathrm{Fix}(T)$.*

Next we discuss the general convex feasibility problem which involves more than two sets. From now on bold symbols are used for sets and operators in the appropriate product space.

Proposition 2.6 *Let $C_1, \ldots, C_r \subset \mathbb{R}^n$ be non-empty, closed and convex sets. Denote $\mathcal{C} := C_1 \times C_2 \ldots \times C_r$ then the projection in the product space is*

$$\boldsymbol{P}_{\mathcal{C}}\left(x^1, \ldots, x^r\right) = \left(P_{C_1}\left(x^1\right), \ldots, P_{C_r}\left(x^r\right)\right) \tag{2.16}$$

and the reflection is

$$\boldsymbol{R}_{\mathcal{C}}\left(x^1, \ldots, x^r\right) = \left(2P_{C_1}\left(x^1\right) - x^1, \ldots, 2P_{C_r}\left(x^r\right) - x^r\right) \tag{2.17}$$

In particular the projection and reflection with respect to the **`diagonal set`**

$$\boldsymbol{D} := \left\{\left(x^1, \ldots, x^r\right) \in \mathbb{R}^{r \cdot n} \mid x^1 = x^2 = \ldots = x^r\right\}$$

is

$$\boldsymbol{P_D}\left(x^1,\ldots,x^r\right)=(\bar{x},\ldots,\bar{x}) \tag{2.18}$$

and

$$\boldsymbol{R_D}\left(x^1,\ldots,x^r\right)=\left(2\bar{x}-x^1,\ldots,2\bar{x}-x^r\right) \tag{2.19}$$

where $\bar{x}=1/r\sum_{i=1}^{r}x^i$.

For the proof of the formulas above we make use of some known facts as well as, for example, [1, Proposition 3.1].

Proof. Given a point $\boldsymbol{x}:=(x^1,\ldots,x^r)\in(\mathbb{R}^n)^r$ and its projection onto the diagonal set $\boldsymbol{P_D}(\boldsymbol{x})=(p,\ldots,p)$. For any $z\in\mathbb{R}^n$, $(z,\ldots,z)\in\boldsymbol{D}$. So

$$0=\langle\boldsymbol{x}-(p,\ldots,p),(z,\ldots,z)\rangle=\sum_{i=1}^{r}\langle x^i-p,z\rangle=\left\langle\sum_{i=1}^{r}x^i-rp,z\right\rangle \tag{2.20}$$

where $p=1/r\sum x^i$ and this complete the proof onto the diagonal set $\boldsymbol{D}$.

For the projection onto the set $\mathcal{C}$, take any $\boldsymbol{c}:=(c^1,\ldots,c^r)\in\mathcal{C}$ and $\boldsymbol{p}:=(p^1,\ldots,p^r)\in\prod_{i=1}^{r}P_{C_i}(x^i)\subseteq\mathcal{C}$

$$\|\boldsymbol{x}-\boldsymbol{c}\|^2=\sum_{i=1}^{r}\|x^i-c^i\|^2\geq\sum_{i=1}^{r}\|x^i-p^i\|^2=\|\boldsymbol{x}-\boldsymbol{p}\|^2 \tag{2.21}$$

and hence $\prod_{i=1}^{r}P_{C_i}(x^i)\subseteq\boldsymbol{P}_{\mathcal{C}}(\boldsymbol{x})$.

On the other hand, for the point $\boldsymbol{p}$ as above, suppose that for some $j\in\{1,\cdots,r\}$, $p^j\notin P_{C_j}(x^j)$. So, for $\boldsymbol{q}:=(q^1,\ldots,q^r)\in(\mathbb{R}^n)^r$ such that $q^j\in P_{C_j}(x^j)$ and $q^i=p^i$ for $i\neq j$. Hence

$$\|\boldsymbol{x}-\boldsymbol{c}\|^2=\sum_{i=1}^{r}\|x^i-p^i\|^2>\sum_{i=1}^{r}\|x^i-q^i\|^2=\|\boldsymbol{x}-\boldsymbol{q}\|^2. \tag{2.22}$$

Since $\boldsymbol{q}\in\mathcal{C}$, we conclude that $\boldsymbol{p}\notin\boldsymbol{P}_{\mathcal{C}}(\boldsymbol{x})$ and the desired result is obtained. ■

Following the above we obtain a product space reformulation which reduces the general feasibility problem to a two-set CFP.

Claim 2.7 *Let $C_1,\ldots,C_r\subset\mathbb{R}^n$ be non-empty, closed and convex sets. Then the following holds.*

$$x^*\in\cap_{i=1}^{r}C_i\Leftrightarrow(x^*,\ldots,x^*)\in\mathcal{C}\cap\boldsymbol{D}. \tag{2.23}$$

Hence the above allows to implement the Douglas-Rachford algorithm (2.14) and the alternating projections method for solving the general CFP as a two-sets CFP in the appropriate product space and obtain a simultaneous variants of the methods. As an example we present next the general DR method. Choose an arbitrary initial point $x_0 \in \mathbb{R}^n$ and define the initial starting point in the product space $\boldsymbol{x_0} := (x_0, \ldots, x_0) \in \mathbb{R}^{r \cdot n}$. Given the k-th iterate $\boldsymbol{x}_k := (x_k^1, \ldots, x_k^r) \in \mathbb{R}^{r \cdot n}$, the next iterate of the Douglas-Rachford algorithm is calculated as

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{T}_{\boldsymbol{D},\mathcal{C}}\left(\boldsymbol{x}_k\right) = \frac{1}{2}\left(\boldsymbol{R}_{\mathcal{C}}\boldsymbol{R}_{\boldsymbol{D}} + \boldsymbol{I}\right)\left(\boldsymbol{x}_k\right) \\ &= \boldsymbol{P}_{\mathcal{C}}\left(2\boldsymbol{P}_{\boldsymbol{D}}\left(\boldsymbol{x}_k\right) - \boldsymbol{x}_k\right) + \left(\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{D}}\right)\left(\boldsymbol{x}_k\right). \end{aligned} \tag{2.24}$$

It is clear that the sequence converges to a point $\boldsymbol{z}$ such that $\boldsymbol{P}_{\boldsymbol{D}}\left(\boldsymbol{z}\right) \in \mathcal{C} \cap \boldsymbol{D}$. Based on that we can use the following iterative step in the original space. This approach was suggested under the name `DR+Proj` in [1].

$$x^{k+1} = \begin{cases} \boldsymbol{P}_{\boldsymbol{D}}\left(\boldsymbol{T}_{\boldsymbol{D},\mathcal{C}}\left(\boldsymbol{x}_k\right)\right) & \text{if } x \in \{400, 800, 1600, 3200, 6400\} \\ \boldsymbol{T}_{\boldsymbol{D},\mathcal{C}}\left(\boldsymbol{x}_k\right) & \text{else }. \end{cases} \tag{2.25}$$

3 The Unary Resource Constraint problem (URC)

In scheduling, unary resource models a set of non-interruptible activities I which must not overlap in time – once a resource starts process an activity it cannot stop or change the activity until processing of the activity is finished. Each activity $i \in I$ can be restricted by the following limits:

(i) the earliest possible starting time $\mathrm{est}_i \in \mathbb{R}_+$,

(ii) the latest possible completion time $\mathrm{lct}_i \in \mathbb{R}_+$,

(iii) the processing time $p_i \in \mathbb{R}_+$.

A (sub)problem is to find a schedule satisfying all these requirements.
See illustration of 3 activities URC in Figure 4.

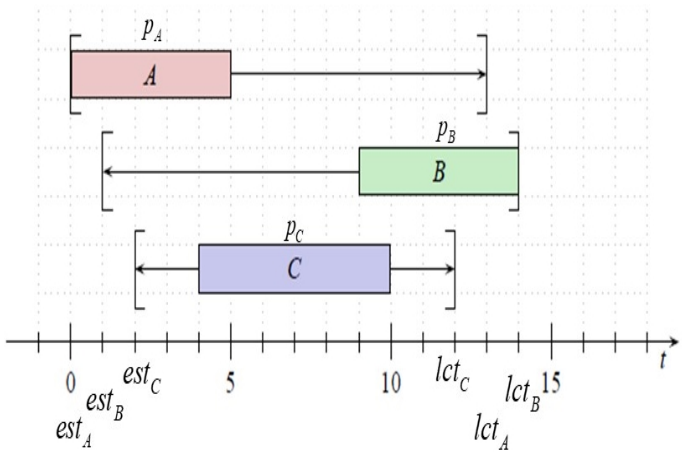


Figure 4: Example of an URC problem with activities denoted by A,B and C.

Next we reformulated the problem as a non-convex feasibility problem.

Notation 3.1 *Given an indices set I, let $x \in \mathbb{R}^I$ with entries* est_i*, that is* $x = (x_1, \ldots, x_I) = (\mathrm{est}_1, \ldots, \mathrm{est}_I)$*. The constraints of URC can be presented as follows.*

(i) ***The processing time constraint****. For any activity $i \in I$ with processing time p_i, the starting time x_i belongs to the interval* $[\mathrm{est}_i, \mathrm{lct}_i - p_i]$*, this is expressed as the following box constraint set*

$$\square := \prod_{i=1}^{N} [\mathrm{est}_i, \mathrm{lct}_i - p_i]. \tag{3.1}$$

(ii) ***Not overlap.*** *For any two activities $i \neq j \in I$ we have*

$$C_{i,j} \cup C_{j,i} := \left\{x \in \mathbb{R}^I \mid x_j - x_i \geq p_i\right\} \cup \left\{x \in \mathbb{R}^I \mid x_i - x_j \geq p_j\right\}. \tag{3.2}$$

This set is non-convex as the union of two half-spaces:

$$C_{i,j} := \left\{x \in \mathbb{R}^I \mid \langle (x_1, \ldots, x_i, \ldots, x_j, \ldots, x_I), (0, \ldots, -1, \ldots, 1, \ldots, 0)\rangle \geq p_i\right\} \tag{3.3}$$

and

$$C_{j,i} := \left\{x \in \mathbb{R}^I \mid \langle (x_1, \ldots, x_i, \ldots, x_j, \ldots, x_I), (0, \ldots, 1, \ldots, -1, \ldots, 0)\rangle \geq p_j\right\}. \tag{3.4}$$

In general one should include a non-negativity constraints, that is

$$\mathbb{R}_+^I := \left\{ x \in \mathbb{R}^I \mid x_i \geq 0 \right\} \tag{3.5}$$

but due to the description of the problem, we can assume that $\square \subset \mathbb{R}_+^I$.

See illustration of the above constraints for two disjoint x_i, x_j in Figures 5-6.

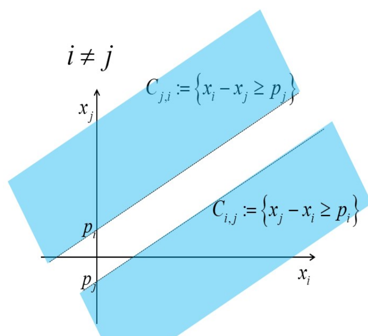


Figure 5: The non-overlapping constraints.

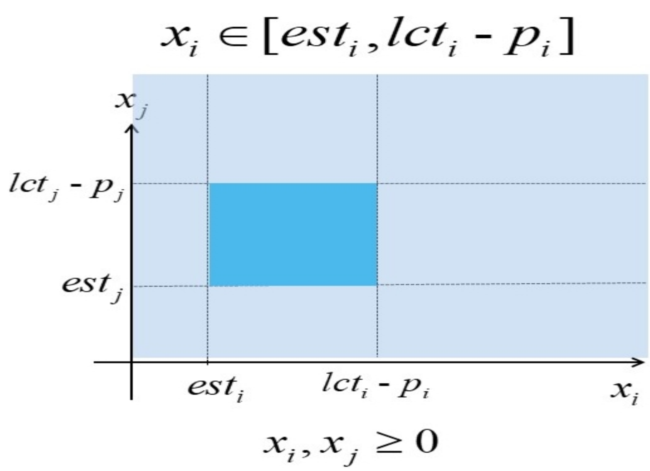


Figure 6: The processing time and non-negativity constraints.

So, solution to the Unary Resource Constraint problem is a solution to the following *non-convex feasibility problem*.

$$\text{Find a point } x^* \text{ such that } x^* \in \square \cap \cap_{i \neq j} C_{i,j}. \tag{3.6}$$

In order to apply Douglas-Rachford algorithm we use the product space transformation and we also have to define the projection mapping onto the constraints sets (3.1) and (3.2). While the projection onto $\square$ is given explicitly by (2.5), respectively; the projection onto (since it is not convex) $C_{i,j} \cup C_{j,i}$ is calculated as the result of the following corollary.

Corollary 3.2 *For any $i, j \in I$ with $i \neq j$, the set $C_{i,j} \cup C_{j,i}$ is non-convex as a union of two non-intersecting half-spaces. For any $x \in \mathbb{R}^I_+$ the projection in $C_{i,j} \cup C_{j,i}$ is given as (not unique).*

$$\left[P_{C_{i,j} \cup C_{j,i}}(x)\right]_{i,j} = \begin{cases} \begin{cases} x_i + (1/2)\,(p_j - \delta_{i,j}) \\ x_j - (1/2)\,(p_j - \delta_{i,j}) \end{cases} & \begin{cases} \text{if } x_i > x_j \\ \text{and } x_i - x_j \leq p_j \end{cases} \\ \\ \begin{cases} x_i - (1/2)\,(p_i - \delta_{i,j}) \\ x_j + (1/2)\,(p_i - \delta_{i,j}) \end{cases} & \begin{cases} \text{if } x_j > x_i \\ \text{and } x_j - x_i \leq p_i \end{cases} \\ \\ x_i,\ x_j & \text{else} \end{cases} \tag{3.7}$$

where $\delta_{i,j} = |x_i - x_j|$.

Proof. *Let $x \in \mathbb{R}^I$ and $i \neq j \in I$. According to (3.7) $P_{C_{i,j} \cup C_{j,i}}(x) = x$. On the other hand if $x \notin C_{i,j} \cup C_{j,i}$ then with out the loss of generality $x_j > x_i$ and $x_j - x_i \leq p_i$ then we calculate the projection of x onto $C_{i,j}$ and onto $C_{j,i}$ ((3.3) and (3.4)) separately based on (2.3).*

$$P_{C_{i,j}}(x) := \begin{cases} x - \frac{x_j - x_i - p_i}{2}\,(0, \ldots, -1, \ldots, 1, \ldots, 0) & \text{if } x \notin C_{i,j} \\ x & \text{if } x \in C_{i,j} \end{cases} \tag{3.8}$$

and

$$P_{C_{j,i}}(x) := \begin{cases} x - \frac{x_i - x_j - p_j}{2}\,(0, \ldots, 1, \ldots, -1, \ldots, 0) & \text{if } x \notin C_{j,i} \\ x & \text{if } x \in C_{j,i}. \end{cases} \tag{3.9}$$

Now if $\left\|x - P_{C_{i,j}}(x)\right\| < \left\|x - P_{C_{j,i}}(x)\right\|$ then $P_{C_{i,j} \cup C_{j,i}}(x) = P_{C_{i,j}}(x)$.and if $\left\|x - P_{C_{i,j}}(x)\right\| > \left\|x - P_{C_{j,i}}(x)\right\|$ then $P_{C_{i,j} \cup C_{j,i}}(x) = P_{C_{j,i}}(x)$. In case that $\left\|x - P_{C_{i,j}}(x)\right\| = \left\|x - P_{C_{j,i}}(x)\right\|$ choose $P_{C_{i,j} \cup C_{j,i}}(x)$ to be either $P_{C_{i,j}}(x)$ or $P_{C_{j,i}}(x)$. This completes the proof. ∎

Remark 3.3 *Let $x \in \mathbb{R}^I$ and denote $\widetilde{x} \in \mathbb{R}^I$ the sorted x in a increasing order, that is $\widetilde{x}_i < \widetilde{x}_{i+1}$ for all $i \in \{1, \ldots, I-1\}$. For $i < j$ with $\tilde{\delta}_{i,j} = |\widetilde{x}_i - \widetilde{x}_j|$ the projection onto $C_{i,j} \cup C_{j,i}$ can be simplified as follows.*

$$\left[P_{C_{i,j} \cup C_{j,i}}(\widetilde{x})\right]_{i,j} = \begin{cases} \begin{cases} \widetilde{x}_i - (1/2)\,(\tilde{p}_i - \delta_{i,j}) \\ \widetilde{x}_j + (1/2)\,(\tilde{p}_i - \delta_{i,j}) \end{cases} & \textit{if } \widetilde{x}_j - \widetilde{x}_i < \tilde{p}_i \\ \\ \widetilde{x}_i,\ \widetilde{x}_j & \textit{else.} \end{cases}$$

The complexity of sequentially evaluating $P_{C_{i,j} \cup C_{j,i}}(\widetilde{x})$ for all $i \neq j$ is then reduces to either $P_{C_{i,j}}(x)$ or $P_{C_{j,i}}(x)$ which are projections onto half-spaces!

Hence, sorting x can be achieved on average of $O(I \log I)$ and then we just have to check the successive coordinates of $\widetilde{x}$ which is $O(I)$, hence $O(I \log I)$ in total. On the other hand, evaluating $P_{\cap_{i \neq j} C_{i,j} \cup C_{j,i}}(x)$ consists of $\binom{I}{2}$ check-ups which is $O(I^2)$.

Another modification which can be used along the run of the algorithm, which also avoids non-convexity, is to check before applying the projection onto $C_{i,j} \cup C_{j,i}$ if $x_i > x_j$ or $x_i \leq x_j$ and then project onto $C_{i,j}$ or $C_{j,i}$ respectively.

Denote $N := \left(1 + \binom{I}{2}\right) = (I(I-1))/2 + 1$. We now transform the URC problem to a feasibility problem in the product space $\mathbb{R}^{I \cdot N}$ with the following two sets

$$\mathcal{C} := \prod_{i=1}^{N} C_i = \square \times \prod_{i \neq j} C_{i,j}, \text{ and the diagonal} \boldsymbol{D}. \tag{3.10}$$

4 The Algorithm

We now present the modified pseudo code of the Douglas-Rachford Algorithm (2.24) for solving the Unary Resource Constraint problem in the product space $\mathbb{R}^{I \cdot N}$ with the sets $\mathcal{C}$ and $\boldsymbol{D}$. Given an indices set I, and consider the product space $\mathbb{R}^{I \cdot N}$.

$$\boldsymbol{x}_{k+1} = \boldsymbol{T}_{\boldsymbol{D},\mathcal{C}}\left(\boldsymbol{x}_k\right) = \frac{\boldsymbol{R}_{\mathcal{C}} \boldsymbol{R}_{\boldsymbol{D}} + \boldsymbol{I}}{2} \left(\boldsymbol{x}_k\right). \tag{4.1}$$

Claim 4.1 *Assume that* est_i, lct_i *and p_i are natural numbers and the URC problem (3.6) is solvable. If the iterative step (4.1) generated a solution*

$\left(x^1, \ldots, x^N\right) \in \mathbb{R}^{I \cdot N}$, *then it can be transformed into an integer solution* $\left(z^1, \ldots, z^N\right) \in \mathbb{N}^{I \cdot N}$.

Proof. Given $\left(x^1, \ldots, x^N\right) \in \mathbb{R}^{I \cdot N}$ we sort it $(O(N \log N))$ in increasing order and obtain $(x^{i_1}, \ldots, x^{i_N})$ such that $i_k \in \{1, \ldots, N\}$ and $x^{i_k} < x^{i_{k+1}}$ for all $k \in \{1, \ldots, N-1\}$. Since we assumed that $\left(x^1, \ldots, x^N\right) \in \mathbb{R}_+^{I \cdot N}$ solves (3.6) it also satisfies the constraints:

$$x^{i_k} \in [\mathrm{est}_{i_k}, \mathrm{lct}_{i_k} - p_{i_k}] \text{ for all } i_k \in \{1, \ldots, N\}, \tag{4.6}$$

and

$$x^{i_k} + p_{i_k} \leq x^{i_{k+1}} \text{ for all } i_k \in \{1, \ldots, N-1\}. \tag{4.7}$$

We claim that $\left(z^1, \ldots, z^N\right) = (\lfloor x^{i_1} \rceil, \ldots, \lfloor x^{i_N} \rceil) \in \mathbb{N}^{I \cdot N}$ also fulfils (4.6) and (4.7). $\lfloor \rceil$ is the closest integer function. If $z^k = \lfloor x^{i_k} \rceil$ for all $i_k \in \{1, \ldots, N\}$ then (4.6) holds; Otherwise there exists at least one $z^k \neq \lfloor x^{i_k} \rceil$. For that i_k we have $\mathrm{est}_{i_k} \leq x_{i_k} \leq \mathrm{lct}_{i_k} - p_{i_k}$ with est_{i_k}, lct_{i_k} and p_{i_k} natural numbers. Hence to the property of the floor function $\mathrm{est}_{i_k} \leq \lfloor x^{i_k} \rceil \leq \mathrm{lct}_{i_k} - p_{i_k}$. For (4.7) we relay on monotonicity of the floor function to obtain for all $i_k \in \{1, \ldots, N-1\}$

$$\begin{gathered} x^{i_k} + p_{i_k} \leq x^{i_{k+1}} \\ \Leftrightarrow \\ z^k + p_{i_k} = \left\lfloor x^{i_k} \right\rceil + p_{i_k} = \left\lfloor x^{i_k} + p_{i_k} \right\rceil \leq \left\lfloor x^{i_{k+1}} \right\rceil = z^{k+1} \end{gathered} \tag{4.8}$$

which completes the proof. ∎

The above result suggest one of the below stopping rule for DR algorithm for solving the URC problem:

$$\lfloor (\boldsymbol{P}_{\boldsymbol{D}}(\boldsymbol{x}_k)) \rceil \in \mathcal{C} \cap \boldsymbol{D} \tag{4.9}$$

or due to the nature of the problem

$$\boldsymbol{P}_{\boldsymbol{D}}(\boldsymbol{x}_k) \in \mathcal{C} \cap \boldsymbol{D}. \tag{4.10}$$

Similar stopping rule was used in [1] for solving for example Sudoku puzzles.

5 Examples

In this section we illustrate two and three dimensional URC examples with the geometric interpretation. Given the following data.

Table 1: The data.

Tasks	est	lst	p	feasible interval
1	1	15	6	$[1, 9]$
2	3	8	3	$[3, 5]$

In Figures 7-9 the constraints are present as well as the solution region.

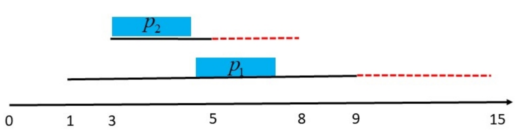


Figure 7: Representation of the data of Table 1.

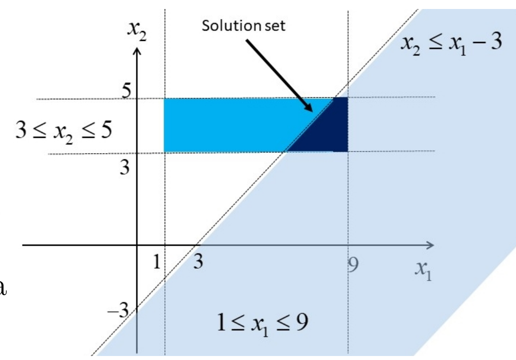


Figure 8: Geometric interpretation of the URC problem given in Table 1.

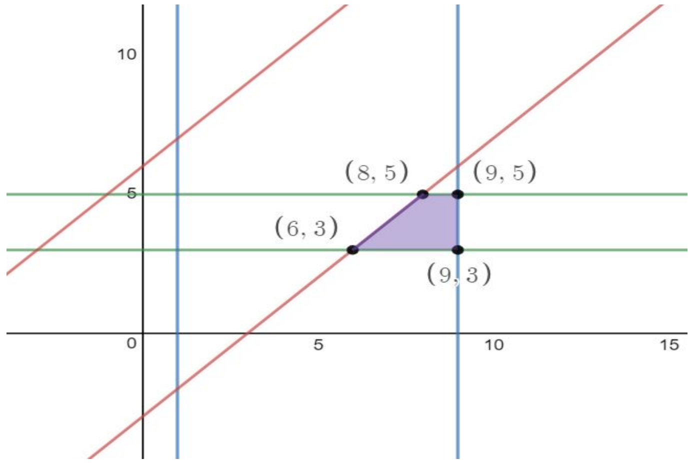


Figure 9: Close-up of the solution region in Figure 8.

Now we consider a three dimensional problem.

Table 2: The data.

Tasks	est	lst	p	feasible interval
1	1	23	5	$[1, 18]$
2	6	18	3	$[6, 15]$
3	2	12	4	$[2, 8]$

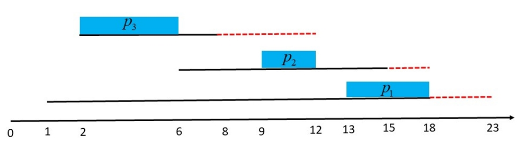


Figure 10: Representation of the data of Table 2.

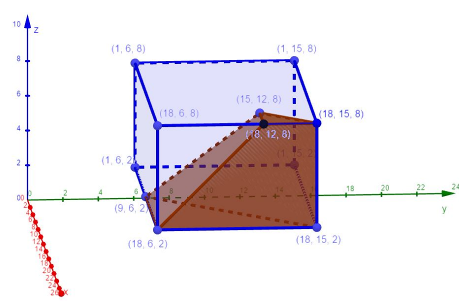
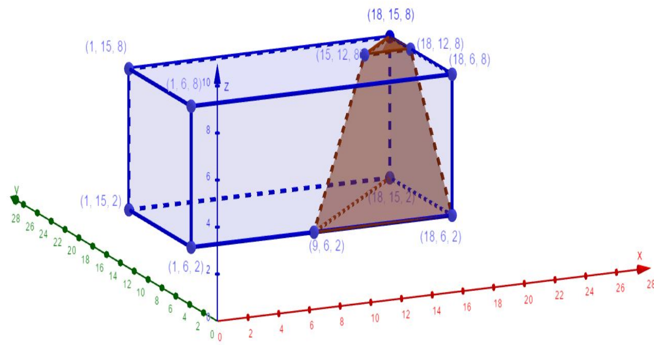


Figure 11: Geometric interpretation of the URC problem given in Table 2.

Figure 12: Another look at the solution region of the URC problem given in Table 2



6 Conclusions

This present project we propose a new mathematical reformulation for a special scheduling problem known as the Unary Resource Constraint problem.While traditionally this is a discrete optimization problem which is solved via methods designed for such cases, the new convex feasibility modelling "open the door" to apply continues optimization methods such as projection methods and in particular Douglas-Rachford and Von Neumann Alternating Projections Algorithms which attracts much attention in recent years due to their effectiveness.

Besides the new reformulation ,this work suggest several interesting direction s for future work, for example instead of finding any solution to the URC, one can apply the newly introduced Superiorization methodology (http://math.haifa.ac.il/YAIR/bib-superiorization-censor.html) which aims for finding a feasible solution which is superior with respect to an additional objective. Clearly one superior solution will be the one with total processing time, but one can also be interested in solutions which terminates the earliest and so.

References

[1] F. J. Aragón Artacho, J. M. Borwein and M. K. Tam, Recent results on Douglas-Rachford methods for combinatorial optimization problems, *Journal of Optimizaion Theory and Applications* **163** (2014), 1–30.

[2] F. J. Aragón Artacho, J. M. Borwein and M. K. Tam, Global behavior of the Douglas-Rachford method for a nonconvex feasibility problem, *Journal of Global Optimization* **65** (2016), 309–327.

[3] F. J. Aragón Artacho, Y. Censor and A. Gibali, The Cyclic Douglas-Rachford algorithm with $\boldsymbol{r}$-sets-Douglas-Rachford Operators, accepted for publication in *Optimization Methods and Software*, 2018.

[4] H. H. Bauschke and J. M. Borwein, On projection algorithms for solving convex feasibility problems, *SIAM Review* **38** (1996), 367–426.

[5] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Second Edition, Springer International Publishing AG, 2017.

[6] H. H. Bauschke, P. L. Combettes, and D. R. Luke, Finding best approximation pairs relative to two closed convex sets in Hilbert spaces, *J. Approx. Theory*, **127** (2004), 178–192.

[7] H. H. Bauschke, M. N. Dao and S. B. Lindstrom, The Douglas–Rachford algorithm for a hyperplane and a doubleton. *Technical report*, April 2018.

[8] H. H. Bauschke and W. M. Moursi, On the Douglas-Rachford algorithm, *Mathematical Programming, Series A* **164** (2017), 263–284.

[9] A. Cegielski, *Iterative Methods for Fixed Point Problems in Hilbert Spaces*, Lecture Notes in Mathematics, vol. **2057**, Springer-Verlag, Berlin, Heidelberg, Germany (2012).

[10] Y. Censor, T. Elfving and G. T. Herman, Averaging strings of sequential iterations for convex feasibility problems, in: Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, D. Butnariu, Y. Censor and S. Reich (Editors), Elsevier Science Publishers, Amsterdam, The Netherlands, 2001, pp. 101-114.

[11] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*, Oxford University Press, New York, NY, USA, 1997.

[12] J. Douglas and H. H. Rachford, On the numerical solution of the heat conduction problem in two and three space variables, *Transactions of the American Mathematical Society* **82** (1956), 421–439.

[13] D. R. C. García, *Contributions to the Theory and Applications of Projection Algorithms*, PhD. thesis, December 2018.

[14] K. Goebel and S. Reich, *Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings*, Marcel Dekker, New York and Basel, 1984.

[15] P. L. Lions and B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.* **16**(1979), 964–979.

[16] J. von Neumann, On rings of operators. Reduction theory, *Annals of Mathematics* **50** (1949), 401–485.

[17] G. Pierra, Decomposition through formalization in a product space, *Mathematical Programming* **28** (1984), 96–115.

Algorithm 2 The Douglas-Rachford Algorithm for solving the URC

1: `Choose an arbitrary starting point` $\boldsymbol{x}_0 = (x_0^1, \cdots, x_0^N) \in \mathbb{R}^{I \times N}$, `where each` $x_0^l \in \mathbb{R}^I$ `for` $l = 1, \cdots, N$.

2: `(Diagonal reflection): Given the current iterate` $\boldsymbol{x}_k = (x_k^1, \cdots, x_k^N)$, `update the point` $\boldsymbol{y}_{k+1} = (y_{k+1}^1, \cdots, y_{k+1}^N)$ `such that for all` $l = 1, \cdots, N$:

$$y_{k+1}^l = \frac{2}{N} \sum_{l=1}^{N} x_k^l - x_k^l. \tag{4.2}$$

3: `(Box reflection): update` $y_{k+1}^1 = \boldsymbol{R}_\square(y_{k+1}^1)$ `using (2.5)`.

4: `(Non-overlap reflection): for any two indices` $i, j \in I$ `(in an increasing order), update the points` $y_{k+1}^2, \cdots, y_{k+1}^N$ `in an increasing order such that for all` $l = 2, \cdots, N$:

5: **if** $[y_{k+1}^l]_i > [y_{k+1}^l]_j$ **then**

6:
$$y_{k+1}^l = 2\boldsymbol{P}_{C_{j,i}}\left(y_{k+1}^l\right) - y_{k+1}^l \tag{4.3}$$
using (3.8).

7: **else if** $[y_{k+1}^l]_j > [y_{k+1}^l]_i$ **then**

8:
$$y_{k+1}^l = 2\boldsymbol{P}_{C_{i,j}}\left(y_{k+1}^l\right) - y_{k+1}^l \tag{4.4}$$
using (3.9)

9: **end if**

10: `(Next iterate): update` $\boldsymbol{x}_{k+1} = (x_{k+1}^1, \cdots, x_{k+1}^N)$ `such that for all` $l = 1, \cdots, N$,

$$x_{k+1}^l = \frac{y_{k+1}^l + x_k^l}{2}. \tag{4.5}$$

11: $k \to k + 1$