Modell für Scheduling

Benötigte Daten

Beschreibung

Die Daten sind in 2 Teile aufgeteilt:

- System-Info, beinhaltet Informationen über die Umgebung in der produziert wird
- Orders, beinhaltet Kundenaufträge/Bestellungen, die in dem beschriebenen System umgesetzt werden sollen

System-Info

Beinhaltet Informationen über die Produktions-Umgebung

Tasks Alle Arbeitsschritte / Aktionen, die in der Produktions-Umgebung durchgeführt werden können. (Wichtig: Tasks die z.Bsp. bei Task A als „follow up task“ angegeben werden sollen KEINEN Verweis auf Task A in den „preciding tasks“ haben).

- id – eindeutige ID des Tasks (kann fortlaufende Nummer sein, nur für interne Identifizierung)
- name – Name des Tasks (für einfacher lesbare Ergebnisse)
- resources – (liste) benötigt Ressourcen, um den Task durchzuführen (id + anzahl)
- result_resources – (liste) alle Ressourcen (ID + Anzahl (Losgröße)) die durch den Task entstehen
- preceding_tasks – (liste) Tasks, die gemeinsam mit diesem Task durchgeführt werden müssen (vorangestellte Tasks, müssen daher nicht extra als Arbeitsschritt übermittelt werden)
- follow_up_tasks – (liste) Tasks, die nach dem Task durchgeführt werden müssen (wie preciding tasks)
- independent – boolean, um festzustellen, ob der Task im Schedule verwendet werden soll (e.g. false für Tasks die immer nur als vorangestellter Task vorkommen und allein keinen Sinn ergeben)
- prepare_time – Zeit die zum Einrichten der Maschine/Arbeitsstation für diesen Task benötigt wird

- unpreprare_time – Zeit, die nach der Verwendung der Maschine/Arbeitsstation für diesen Task benötigt wird

Recipes Die Rezepte für die verschiedenen Ressourcen, die hergestellt werden sollen (Produkte oder auch Zwischenprodukte)

- id – eindeutige ID des Rezepts (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- name – Name des Rezepts (für einfacher lesbare Ergebnisse)
- tasks – (liste) IDs der Tasks, die durchgeführt werden müssen, um das Rezept umzusetzen (Ergebnis und Ergebnismenge ergeben sich aus dem letzten Task)

Resources Alle Ressourcen, die in der Produktions-Umgebung verwendet oder hergestellt werden können

- id – eindeutige ID der Ressource (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- name – Name der Ressource (für einfacher lesbare Ergebnisse)
- stock – Anzahl der Ressource im Lagerbestand
- price – Preis pro Einheit, um die Ressource einzukaufen (0 wenn die Ressource nicht gekauft werden kann)
- renewable – boolean, um festzustellen, ob die Ressource verbraucht wird oder nicht (könnte z.Bsp. auch Ressource Mitarbeiter sein, der wieder verfügbar ist, wenn ein Task beendet ist)
- recipes – (liste) IDs der Rezepte, durch die diese Ressource hergestellt werden kann (falls vorhanden, sonst leere Liste)

Workstations Alle Maschinen / Arbeitsstationen, die in der Produktions-Umgebung vorhanden sind

- id – eindeutige ID der Maschine/Arbeitsstation (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)
- name – Name der Maschine/Arbeitsstation (für einfacher lesbare Ergebnisse)

- basic_resources – (liste) beinhaltet die IDs der benötigten Ressourcen + die Anzahl für jede Ressource, die (unabhängig der bevorstehenden Tasks) benötigt werden um die Maschine/Arbeitsstation zu betreiben (z.Bsp. Mitarbeiter)

- tasks – (liste) IDs aller Tasks, die auf dieser Maschine/Arbeitsstation durchgeführt werden können + die Dauer des Tasks auf dieser Maschine/Arbeitsstation

Orders

Repräsentiert Kundenaufträge/Bestellungen, die mithilfe der Produktions-Umgebung umgesetzt werden soll.

- id – eindeutige ID der Bestellung (kann auch fortlaufende Nummer sein, nur für interne Identifizierung)

- arrival_time – Zeitpunkt, zu dem die Bestellung im System eingegangen ist

- delivery_time – Zeitpunkt, zu dem die Bestellung fertig sein soll

- latest_acceptable_time – Spätester Zeitpunkt, zu dem die Bestellung fertig sein soll, falls der gewünschte Lieferzeitpunkt nicht eingehalten kann

- resources – (liste) IDs + Anzahl der bestellten Ressourcen, die hergestellt werden sollen, + der Preis, den der Kunde für diese Ressource bezahlen wird

- penalty – falls die Bestellung nicht durchgeführt wird

- tardiness_fee – falls die Bestellung nach dem gewünschten Termin, aber vor dem letzten möglichen Termin durchgeführt wird

- divisible – boolean, gibt an ob auch nur Teile der Bestellung bearbeitet werden können, oder ob alle bestellten Ressourcen geliefert werden müssen (oder gar keine)

- customer_id – ID des Kunden, von dem der Auftrag stammt

Input / Output

Genaue Struktur der Datenübertragung mit Beispieldaten unter "JSON-Beispiel"(section)

Input:
Orders:
$OrderA$
$Task_1 - Task_n$
$price_1 - price_n$
$penalty$
$tardiness\_fee$
$delivery\_date$
$latest\_date$

Mapping von Tasks zu Jobs Bsp.:
(Zuteilung zu exakten Job, Bsp. Task: Material schneiden -> Job: Material schneiden für Bestellung A)
$TaskA.1$
$TaskID$
$JobID$

Intermediate:
$j_1 - (w, s)$
$j_2 - (w, s)$
$j_2 - (w, s)$
.
.
.
$j_n - (w, s)$

Output:
$$\begin{bmatrix} (j_{11}, s_{11}) & (j_{12}, s_{12}) & (j_{13}, s_{13}) & \dots & (j_{1n}, s_{1n}) \\ (j_{21}, s_{21}) & (j_{22}, s_{22}) & (j_{23}, s_{23}) & \dots & (j_{2n}, s_{2n}) \\ \dots & \dots & \dots & \dots & \dots \\ (j_{m1}, s_{m1}) & (j_{m2}, s_{m2}) & (j_{m3}, s_{m3}) & \dots & (j_{mn}, s_{mn}) \end{bmatrix}$$ Dauer?

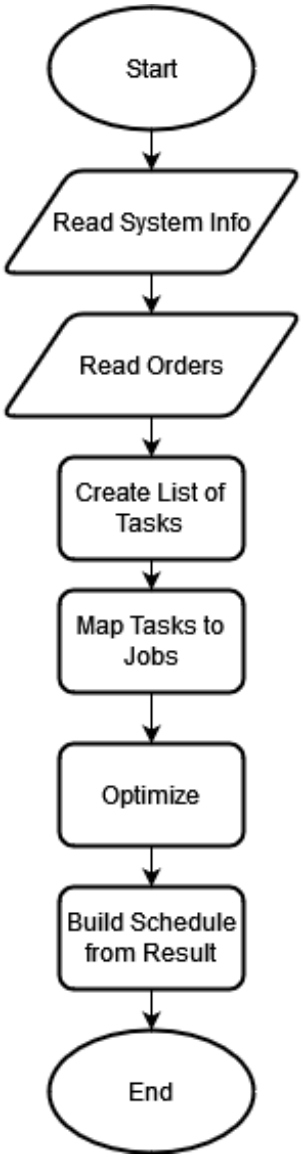

Figure 0.1: Ablauf

Modell

Variablen Definition

O - Set of all Orders
o - Specific Order, $o \in O$
W - Set of all Workstations
T - Set of all Tasks

T_o - Set of all Tasks needed for Order o, $T_o \subset T$
x - Specific Task, $x \in T$
T_{xp} - Set of all Tasks preceding Task x, $T_{xp} \subset T$
T_{xf} - Set of all Tasks following Task x, $T_{xf} \subset T$
J - Set of all Jobs
J_w - Set of all Jobs assigned to Workstation w, $w \in W$
j_x - Job linked to Task j, $x \in T$, $j \in J$
x_j - Task linked to Job j, $x \in T$, $j \in J$
W_j - Set of Workstations eligible for Job j, $W_j \subset W$
R - Set of all Resources
w - Specific Workstation $w \in W$
R_j - Set of Resources needed for Job j, $R_j \subset R$
r - Specific Resource, $r \in R$
d_{xw} - Duration of Task x on Workstation w
j_w - Selected Workstation for a Job j, $w \in W_j$
j_s - Start time slot of Job j
j_{dt} - Deliver time slot of Job j
j_{lt} - Latest allowed time slot of Job j
i_{rt} - Amount of Resource r in the Inventory at time slot t
l_j - Binary Variable, 1 if Job j is late (Tardiness Fee applies)
l_o - Binary Variable, 1 if Order o is late (Tardiness Fee applies)
u_o - Binary Variable, 1 if Order o can only be fulfilled partially
p_o - Price the customer pays for Order o
f_o - Price reduction for tardy orders
O_c - Set of all Orders which could be completed in the derived schedule, $O_c \subset O$

Objective Function + Nebenbedingungen

Objective Functions

Equation 1 Minimize tardy jobs
Equation 2 Maximize earning
Equation 3 Minimize deviation from the expected delivery date

$$j := J_i$$
$$minimize \sum_{i=0}^{J} l_j \tag{1}$$

$$
\begin{aligned}
o &:= O_{c,i} \\
maximize \sum_{i=0}^{O_c} p_o - (l_o * f_o)
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
j &:= J_i \\
y &:= x_j \\
minimize \sum_{i=0}^{J} |(j_s + d_{y,w}) - j_{dt}|
\end{aligned}
\tag{3}
$$

Nebenbedingungen

Equation 4 to make sure jobs finish before last possible time slot
Equation 5 to make sure jobs have sufficient resources to start
Equation 6 checks if tardiness fee applies
Equation 7 jobs can't start before preceding tasks finish
Equation 8 follow up jobs can't start before job finished
Equation 9 only one active job at one time slot for each Workstation
Equation 10 starting times need to be after timeslot 0

$$
j_s \leq j_{lt} - d_{j,w} \tag{4}
$$

$$
\sum_{m=0}^{R_j} (i_{r_m,t} - r_m \leq 0) < 1 \tag{5}
$$

$$
l_j := j_{dt} \leq j_s \leq j_{lt} \tag{6}
$$

$$
\begin{aligned}
y &\in T_{j_x p} \\
\sum_{m=1}^{y} ((j_{y_m,s} + d_{y_m,w}) - (j_{y_{m-1}} + d_{y_{m-1},w}) < 0) < 1
\end{aligned}
\tag{7}
$$

$$y \in T_{j_x f}$$
$$\sum_{m=1}^{y}((j_{y_{m-1},s} + d_{y_{m-1},w}) < j_{y_m,s}) < 1 \tag{8}$$

$$\begin{aligned} \forall w \in W \\ i = J_{wm} \\ j = J_{wn} \\ x = x_i \\ x' = x_j \end{aligned} \tag{9}$$
$$\sum_{m=0}^{J_w}(\sum_{n=0}^{J_w-1}((1 - (j_s + d_{x',w} < i_s)) + (1 - (j_s > i_s + d_{x,w})))) < 1$$

$$j = J_m$$
$$\sum_{m=0}^{J}(j_s < 0) < 1 \tag{10}$$

JSON-Beispiel

```
{
"system-info":
{
        "tasks":
        [
                {
                "id": 0,
                "name": "example_task",
                "resources":
                [
                        {
                        "id": 4,
                        "amount": 20
                        },
                        {
                        "id": 6,
```

```
                        "amount": 2
                        }
                ],
                "result_resources":
                [
                        {
                        "resource_id": 1,
                        "amount": 100
                        },
                        {
                        "resource_id": 2,
                        "amount": 10
                        }
                ],
                "preceding_tasks":
                [
                        1, 2, 3
                ],
                "follow_up_tasks":
                [ ],
                "independent": true,
                "prepare_time": 10,
                "unprepare_time": 5
                },
                {
                "id": 1,
                "name": "example_task 2",
                "resources":
                [
                        {
                        "id": 7,
                        "amount": 4
                        }
                ],
                "result_resources":
                [
                        {
                        "resource_id": 3,
                        "amount": 4
                        }
```

```
                ],
                "preceding-tasks":
                [ ],
                "follow_up_tasks":
                [
                        4, 5
                ],
                "independent": false,
                "prepare_time": 10,
                "unprepare_time": 5
                }
        ],
        "recipes":
        [
                {
                        "id": 0,
                        "name": "example_recipe",
                        "tasks":
                        [
                                0, 4
                        ]
                },
                {
                        "id": 1,
                        "name": "example_recipe 2",
                        "tasks":
                        [
                                5, 6
                        ]
                }
        ],
        "resources":
        [
                {
                        "id": 0,
                        "name": "example_resource",
                        "stock": 500,
                        "price": 10,
                        "renewable": false,
                        "recipes":
```

```
                        [
                                0, 1
                        ]
                },
                {
                        "id": 1,
                        "name": "example_resource 2",
                        "stock": 300,
                        "price": 15,
                        "renewable": true,
                        "recipes":
                        [ ]
                }
        ],
        "workstations":
        [
                {
                        "id": 0,
                        "name": "example_machine",
                        "basic_resources":
                        [
                        {
                                "id": 10,
                                "amount": 5
                        },
                        {
                                "id": 12,
                                "amount": 3
                        }
                        ],
                        "tasks":
                        [
                                {
                                        "task_id": 0,
                                        "duration": 10
                                },
                                {
                                        "task_id": 3,
                                        "duration": 5
                                },
```

```
                                        {
                                                "task_id": 4,
                                                "duration": 20
                                        }
                                ]
                        }
                ]
},
"orders":
[
                {
                        "id": 0,
                        "arrival_time": "07.02.2022 00:00:00",
                        "delivery_time": "20.02.2022 14:30:00"
                           ,
                        "latest_acceptable_time": "26.02.2022
                           00:00:00",
                        "resources":
                        [
                                {
                                        "id": 1,
                                        "amount": 20,
                                        "price": 700
                                },
                                {
                                        "id": 3,
                                        "amount": 10,
                                        "price": 300
                                }
                        ],
                        "penalty": 100,
                        "tardiness_fee": 10,
                        "divisible": true,
                        "customer_id": 1
                },
                {
                        "id": 0,
                        "arrival_time": "09.02.2022 13:00:00",
                        "delivery_time": "25.02.2022 15:00:00"
                           ,
```

```
                "latest_acceptable_time": "26.02.2022
                   00:00:00",
                "resources":
                [
                        {
                                "id": 0,
                                "amount": 15,
                                "price": 500
                        },
                        {
                                "id": 1,
                                "amount": 30,
                                "price": 1000
                        },
                        {
                                "id": 2,
                                "amount": 10,
                                "price": 200
                        }
                ],
                "penalty": 100,
                "tardiness_fee": 10,
                "divisible": false,
                "customer_id": 0
        }
]
}
```