Resource-constrained scheduling with non-constant capacity and non-regular activities[1]

Giorgio Fasano

Thales Alenia Space Italia S.p.A., Str. Ant. di Collegno 253, 10146, Turin, Italy
giorgio.fasano@thalesaleniaspace.com

Abstract: This work is inspired by very challenging issues arising in space logistics. The problem of scheduling a number of activities, in a given time elapse, optimizing the resource exploitation is discussed. The available resources are not constant, as well as the request, relative to each job. The mathematical aspects are illustrated, providing a time-indexed MILP model. The case of a single resource is analysed first. Extensions, including the multi-resource case and the presence of additional conditions are considered. Possible applications are suggested and an in-depth experimental analysis is reported.

Keywords: *resource constrained project scheduling problem (RCPSP), non-constant resource capacity, non-constant resource request, irregular job/activity/cycle profile, multi-resource, time-indexed scheduling, mixed integer linear programming (MILP), global optimization (GO)*

1 Introduction

This work is inspired by the logistic context in space activities. It is notorious that, in this framework, the exploitation of the resources available (e.g. on orbit or on the exploration surface) is usually an extremely challenging issue. Complex scheduling problems arise, presenting the experts with the necessity of optimizing the sequencing of what is usually a significant number of jobs, requiring contemporarily the utilization of different resources, such as, electrical power, data handling capacity and crew time. As a further non-trivial difficulty, the operational cycles (jobs) are frequently associated with an irregular activity, i.e. they are characterized by a variable resource request profile. Similarly, the overall capacities of the relevant resources vary. Fig. 1 provides, as an illustrative example, the case of a single (non-constant) resource and three different (non-constant) request cycle types.

Fig. 1 Irregular cycles and non-constant resource

The specialist literature on scheduling is vast [1,3,5,6,11,8-20,22,26,28,31], covering several specific problems and methodologies. This chapter focuses on a non-standard resource constrained project scheduling problem (RCPSP). For the classical RCPSP see [4,8-10,13,15,17,23,25,29,30,32]. In the author's previous work [12], an approximate MILP (Mixed Integer Linear Programming) time-continuous approach, was proposed. A novel approximate MILP formulation of the problem, based on a time-indexed approach, is discussed here. This latter option was motivated by the efficiency of discretized models for scheduling problems [2]. As in the previous work, the approach proposed in this chapter provides a global optimization (GO) perspective on the problem in question. The discussed formulation is suitable for tackling a number of different variants of the RCPSP, involving either single or multiple resources and characterized by the specific objective functions adopted.

The remainder of the chapter is organized as follows. The first part of Section 2 provides an MILP model for the case of a single resource, namely electrical power [11,21,24]. Afterwards, the presence of possible additional conditions is outlined and an extended formulation, addressing the multi-resource scenario is introduced. Section 3 is devoted to the computational study.

[1] This is a draft version of the chapter 'Resource-constrained scheduling with non-constant capacity and non-regular activities' to appear in: Giorgio Fasano and János D. Pintér, Eds. Space Engineering: Modeling and Optimization with Case Studies. Springer Science + Business Media, New York, 2016. Some parts of the text / figures have been removed.

2 Time-indexed formulation

The problem considered in this section, concerns the electrical power consumption, by a number of devices (e.g. payloads, in the case of the space framework), in a pre-specified time period. Each device may be requested to execute a sequence of cycles, between a given minimum and maximum limit, i.e. $\underline{N}_\tau$ and $\overline{N}_\tau$, respectively. Assuming, for the sake of simplicity, that the value associated with each device cycle is the same (this assumption could be generalized by introducing appropriate weights), the optimization objective consists of maximizing the exploitation of the energy available during the entire time period $[0,T_f]$, where T_f denotes the final time.

Some additional conditions (e.g. of time-precedence), might be imposed on the execution of the cycles (relevant examples shall be provided). The (electrical) power available at each instant $t \in [0,T_f]$ is represented by a given function of time $w(t)$ (e.g. step-wise or continuous, see Fig. 1). Similarly, each cycle type $\tau \in T$ (T indicates the set of all cycle types) is associated with a given function of time (cycle type profile) $w_\tau(t)$, defined (conventionally) over $[0,D_\tau]$, where D_τ corresponds to cycle type τ duration. All activated cycles must obviously be entirely executed within the given overall time period. This means that, denoting the initial instant of cycle i of type τ with $t_{0\tau i}$, $\forall \tau, \forall i$ $t_{0\tau i} \in [0,T_f - D_\tau]$ (in the following I shall indicate the generic set of cycle indices).

For each cycle of each type τ, the binary variables $\eta_{\tau i} \in \{0,1\}$ are introduced with the following meaning:

$\eta_{\tau i} = 1$ if cycle i of type τ is activated;

$\eta_{\tau i} = 0$ otherwise.

For each cycle i of type τ, the function of time $w_{\tau i}(t)$ is defined as follows:

$$\forall t \in [t_{0\tau i}, t_{0\tau i} + D_\tau], \quad w_{\tau i}(t) = w_\tau(t - t_{0\tau i}), \tag{1-1}$$

$$\forall t \notin [t_{0\tau i}, t_{0\tau i} + D_\tau] \quad w_{\tau i}(t) = 0. \tag{1-2}$$

More precisely, this means that each $t_{0\tau i} \in [0,T_f - D_\tau]$ generates a specific $w_{\tau i}(t)$, belonging to the set of functions with compact support (such that $\forall t \in [o,T_f]$ $w_{\tau i}(t) \geq 0$ and $\forall t \notin [o,T_f]$ $w_{\tau i}(t) = 0$). In the following, only their restrictions to the intervals $[0,T_f]$ will be considered.

The optimization task, in a normalized form, can be expressed as follows:

$$\max_{\substack{\eta_{\tau i},\\ t_{0\tau i} \in [0,T_f - D_\tau]}} \frac{\sum\limits_{\substack{\tau \in T\\ i \in I}} \int_0^{T_f} \eta_{\tau i} w_{\tau i}(t)dt}{\int_0^{T_f} w(t)dt}. \tag{2}$$

Here, $\int_0^{T_f} w(t)dt = E$ represents the total energy available, while $\int_0^{D_f} w_\tau(t)dt = E_\tau$ the energy requested by each cycle of type τ (therefore each integral $\int_0^{T_f} \eta_{\tau i} w_{\tau i}(t)dt$ appearing in (2) may simply be substituted with $\eta_{\tau i} E_\tau$).

It is assumed, without any loss of generality, that

$$\forall \tau \in \mathrm{T}, \forall i,j \in I / i < j \quad [t_{0\tau i}, t_{0\tau i} + D_\tau] \cap [t_{0\tau j}, t_{0\tau j} + D_\tau] = \varnothing. \tag{3}$$

Were parallel processes (for the same cycle types) indeed to be considered, it would be sufficient to extend the set T appropriately.

For any selection $(\underline{t}_{011},...,\underline{t}_{0\tau i},...,\underline{t}_{0|T||I|})$, where $|T|$ and $|I|$ represent the cardinalities of T and I respectively, the following conditions are imposed upon the corresponding functions of time $w_{\tau i}(t)$:

$$\forall t \in [0,T_f] \quad \sum_{\substack{\tau\in \mathrm{T}\\ i\in I}} \eta_{\tau i} w_{\tau i}(t) \le w(t). \tag{4}$$

If a minimum and a maximum limit are specified on the number of cycles, the following constraints are added $\forall \tau \in \mathrm{T}$ $\underline{N}_\tau \le \sum_{i\in I} \eta_{\tau i} \le \overline{N}_\tau$. (If a proper set of indices I_τ were defined for each cycle type, all variables $\eta_{\tau i}$ outside the corresponding index ranges could be eliminated from the model, together with the previous upper limit conditions).

The continuous-time model outlined above, although quite simple to formulate, is extremely difficult to solve by an exact approach. A very simple time-indexed reformulation is therefore put forward hereinafter, in order to provide approximate solutions, useful in practice. To this purpose, a discretization of the entire period $[0,T_f]$ (from now on, it is assumed $T_f \in \mathbb{N}$) is carried out, by utilizing an appropriate time unit, i.e.: $[0,T_f] = [0,1] \cup ... \cup [\nu,\nu+1] \cup ... \cup [T_f - 1, T_f]$. The power function associated with each corresponding sub-interval $[\nu,\nu+1]$ is now assumed to be constant. This gives rise to an approximating step-function, whose values W_ν are defined as follows:

$$\forall \nu / 0 \le \nu \le T_f - 1 \qquad W_\nu = \min_{t\in[\nu,\nu+1]} w(t) \tag{5}$$

Analogously, the activity period associated with each cycle type is discretized. To this purpose, each duration D_τ is substituted with a new one, consisting of the shortest integer interval $\overline{D}_\tau$, in terms the above mentioned time unit, containing D_τ (i.e. $\overline{D}_\tau = \lceil D_\tau \rceil$). The sub-intervals $[0,1]$,..., $[\gamma,\gamma+1]$, ...,$[\overline{D}_\tau - 1, \overline{D}_\tau]$ are subsequently associated to each $\overline{D}_\tau$. Also in this case, for each cycle type, the power consumption, corresponding to each sub-interval $[\gamma,\gamma+1]$, is assumed to be constant and the function $w_\tau(t)$ is therefore approximated by a step-function, whose values are now expressed as:

$$\forall \tau, \forall \gamma / 0 \le \gamma \le \overline{D}_\tau - 1 \qquad W_{\tau\gamma} = \max_{t\in[\gamma,\gamma+1]} w_\tau(t) \tag{6}$$

Remark 1 The adopted approximations for the functions $w(t)$ and $w_\tau(t)$ guarantee that every solution of the discretized model is a feasible solution of the time-continuous one.

For each cycle type τ, the time limit $T_{f\tau} = T_f - \overline{D}_\tau$ is stated. It represents the maximum time breakpoint at which such a cycle type can be activated, in order to be entirely executed within the interval $[0,T_f]$. The binary variables $\chi_{\tau\nu} \in \{0,1\}$ are then defined, with the following meaning:

$\chi_{\tau\nu} = 1$ if a cycle of type τ is activated at instant ν, such that $0 \le \nu \le T_{f\tau}$;
$\chi_{\tau\nu} = 0$ otherwise.

A basic formulation of the approximated MILP model reads as follows. Firstly, objective function (2) is transformed into:

$$max \sum_{\substack{\tau \in \mathrm{T} \\ \nu \le T_{f\tau}}} \frac{E_\tau}{E} \chi_{\tau\nu} \ . \tag{7}$$

The constraints below are introduced:

$$\forall \tau, \forall \nu / 0 \le \nu \le T_{f\tau}, \forall \gamma / \nu \le \gamma \le \nu + \overline{D}_\tau - 1 \quad u_{\tau\gamma\nu} = W_{\tau\gamma} \chi_{\tau\nu} \, , \tag{8}$$

$$\forall \gamma / 0 \le \gamma \le T_f - 1 \qquad \sum_{\substack{\tau \in \mathrm{T} \\ \gamma - D_\tau + 1 \le \nu \le \gamma}} u_{\tau\gamma\nu} \le W_\gamma \, , \tag{9}$$

$$\forall \tau \in \mathrm{T}, \forall \nu / 0 \le \nu \le T_{f\tau} \qquad \chi_{\tau\nu} + \sum_{\substack{\nu' \ge \nu \\ \nu' \le \nu + D_\tau - 1}} \chi_{\tau\nu'} \le 1 \, , \tag{10}$$

$$\forall \tau \in \mathrm{T} \qquad \underline{N}_\tau \le \sum_{\nu \le T_{f\tau}} \chi_{\tau\nu} \le \overline{N}_\tau \, , \tag{11}$$

The variables $u_{\tau\gamma\nu}$ (defined a priori as continuous) express, through equations (8), the power consumption associated with a cycle of type τ, during the sub-interval $[\nu + \gamma, \nu + \gamma + 1]$, if activated at instant ν (in such a case the power consumption equals $W_{\tau\gamma}$). If no cycle of type τ is activated at instant ν, the relative variables $u_{\tau\gamma\nu}$ are zero.

Inequalities (8) and (9) state that during each time sub-interval $[\nu, \nu + 1]$ the power request cannot exceed what is available. Conditions (10) prevent the (total or partial) simultaneity of two (or more) cycles of the same type. The minimum and maximum limits for each cycle type are respected in virtue of inequalities (11).

As a first consideration, conditions (8) and (9) could be rewritten in a single one, getting rid of the variables $u_{\tau\gamma\nu}$, i.e.:

$$\forall \gamma / 0 \le \gamma \le T_f - 1 \quad \sum_{\substack{\tau \in \mathrm{T} \\ \gamma - D_\tau + 1 \le \nu \le \gamma}} W_{\tau\gamma} \chi_{\tau\nu} \le W_\gamma \tag{12}$$

Remark 2 Once (8) and (9) are substituted with (12), all the variables are of the binary type only (binary integer programming, BIP, model).

Hereinafter, extensions of the basic discretized model shall be outlined, considering firstly the possibility of including additional conditions. Two relevant examples are illustrated. The first refers to the case where the total number of cycles of type τ' is a multiple of the one of τ''. This is expressed by the following equations:

$$\sum_{\nu \le T_{f\tau'}} \chi_{\tau'\nu} = R_{\tau'\tau''} \sum_{\nu \le T_{f\tau''}} \chi_{\tau''\nu} \, , \tag{13}$$

($R_{\tau'\tau''} \in \mathbb{N}$). A second example contemplates the case where the execution of each cycle of type τ'' must be preceded by (at least) $P_{\tau'\tau''}$ cycles of type τ'. It is understood, in particular, that each activated τ''-cycle can always be associated (through an injective function) with a set of $P_{\tau'\tau''}$ preceding τ'-cycles and all these sets are disjoint. The conditions below serve the scope:

$$\forall \nu / \nu \le T_{f\tau''} \qquad P_{\tau'\tau''} \sum_{\nu'' \le \nu} \chi_{\tau''\nu''} \le \sum_{\nu' \le \nu - D_{\tau'}} \chi_{\tau'\nu'} \tag{14}$$

As mentioned previously, the approach proposed in this work is extendible to the cases where a number of different resources have to be allowed for. The relevant formulation is briefly reported in the following. The symbolism adopted hitherto is adapted to the extended context, in order to stress the analogies with the basic model. To this purpose, the functions associated with the resources available, whose set is denoted by *R*, are now simply indicated with $w_r(t)$, where $r \in R$ is the corresponding index. With an obvious meaning of the symbols, the extended version of the basic model is reformulated as follows, keeping inequalities (10) and (11) unaltered (and corresponding to (17) and (18) below), i.e.:

$$max \sum_{\substack{r\in R \\ \tau\in \mathrm{T} \\ \nu\le T_{f\nu}}} \frac{E_{r\tau}}{|R|E_r} \chi_{\tau\nu} \qquad (15)$$

$$\forall r \in R, \forall \tau \in \mathrm{T}, \forall \gamma / 0 \le \gamma \le T_f - 1 \qquad (16)$$
$$\sum_{\substack{\tau\in \mathrm{T} \\ \nu - D_\tau + 1 \le \gamma \le \nu}} W_{r\tau\gamma} \chi_{\tau\nu} \le W_{r\gamma},$$

$$\forall \tau \in \mathrm{T}, \forall \nu / 0 \le \nu \le T_f - 1 \qquad (17)$$
$$\chi_{\tau\nu} + \sum_{\substack{\nu'\ge\nu \\ \nu'\le\nu+D_\tau-1}} \chi_{\tau\nu'} \le 1,$$

$$\forall \tau \in \mathrm{T} \qquad \underline{N}_\tau \le \sum_{\nu\le T_{f\tau}} \chi_{\tau\nu} \le \overline{N}_\tau. \qquad (18)$$

Different versions of objective function (15) could also be conceived, if necessary, introducing proper weights, depending on the relevance of each single resource. Additional conditions such as those represented by constraints (13) and (14) could moreover be introduced.

The basic MILP model, when expressed by (7) (10), (11) and (12), contains:

$O(|T|T_f)$ binary variables $\chi_{\tau\nu}$;

$O(|T|T_f)$ cycle non-simultaneity constraints;

$O(2|T|)$ cycle minimum and maximum number constraints;

$O(T_f)$ power capacity constraints.

In the multiple-resource case, the relative number of capacity constrains becomes:

$O(|R|T_f)$.

Remark 3 Differently from the usual indexed-packing-like formulations for scheduling, in the models presented here, the generation of binary variables depends solely on the time discretization adopted and the total number of cycle types involved.

3 Applications and computational results

Time-indexed methods for scheduling problems are well known for their efficiency both in terms of solution quality and computational time. This is essentially due to the fact that their LP-relaxations provide, in general, strong bounds. The corresponding matrix size/density, nonetheless, usually represents a major difficulty and, as a consequence, the computer's memory capacity often becomes the actual stumbling block.

The approach proposed in this chapter is addressed to the previously discussed non-standard scheduling problems, bearing in mind a 'reasonable' limitation for the sizes of the instances to cope with. As a rule of thumb, problems with fewer than 250 sub-intervals (time units) and 35 cycle types, involving 3 different resources, are expected to be solved quite easily, as well as equivalent instances, in terms of matrix size.

Obviously, from a practical point of view, a large-scale problem could be subdivided into a number of sub-problems, by partitioning the total time period appropriately. Moreover, the author's time-continuous model [12] may be utilized within a heuristic procedure, to refine the approximated solutions obtained with the time-indexed approach. To this purpose, when a single resource is involved, it is opportune to interpret each discretized cycle as if it were composed of a number of components corresponding to successive sub-periods (whose duration is not necessarily integer) having constant consumption (extensions to the multiple resource case can be considered). Precedence constraints, deriving from the solution found through the time-indexed model, are hence imposed. They assume the form $t_{h\tau} - t_{k\zeta} \geq D_{h\tau} + D_{k\zeta}$, where $t_{h\tau}$ and $t_{k\zeta}$ are the (time) coordinates (with respect to the given time-resource reference frame) of the centers of components *h* and *k* of cycles τ and ζ respectively, while $D_{h\tau}$ and $D_{k\zeta}$ are the corresponding durations. This way, further cycles can tentatively be added by following an overall *hole-filling* logic [12].

Analogies between some classes of scheduling and packing problems (e.g. [14]) are well known. Applications of the approach proposed in this chapter to two-dimensional rectangular packing, when an MILP-based formulation is adopted (e.g. [27]) are quite straightforward. Similarly to the above mentioned time-precedence constraints, relative positions, derived from the discretized model, indeed, can be imposed with respect to one of the axes, in order to solve the overall packing model. This (heuristic) approach is expected to prove quite advantageous as a support strategy to solve hybrid packing models (e.g. [7]).

Hereinafter, a significant number of tests concerning the class of non-standard scheduling problems discussed in the previous section are reported. They are grouped in the following sets: Basic, A, B, C, D, E, F and G. Additionally, considering the analogies between scheduling and packing problems, a set of two-dimensional rectangular packing instances from literature (Fekete and Shepers, see www.or.deis.unibo.it/research_pages/ORinstances/ORinstances; www.fe.up.pt/esicup) have been taken into account. These are denoted as FS. The Basic test set is considered firstly. All other test sets (except for FS) have been constructed as extensions of the Basic set. These test sets are introduced, in this section, step by step.

All the case studies considered have been solved by utilizing IBM ILOG CPLEX 12.3 [16], supported by a personal computer, equipped with: Core 2 Duo P8600, 2.40 GHz processor; 1.93 GB RAM; MS Windows XP Professional, Service Pack 2.

3.1 Basic test set

In all tests of the Basic set, the (electrical) power is chosen (with reference to the formulation of Section 2) as the only resource considered, with a constant capacity of 25 (power) units. Fifty types of cycles have been defined. They are reported in Table 1. For each cycle τ, the term $K_{h\tau} \times L_{h\tau}$ is associated with its component h (see above). $K_{h\tau}$ indicates the (constant) consumption and $L_{h\tau}$ the duration of the corresponding sub-period, i.e. the number of sub-intervals covered by component h, expressed in time units. It is understood that the duration of each sub-interval is one time unit and $L_{h\tau}$ is integer ($K_i L_i$ is the energy requested by component h). In the table, all the terms $K_{h\tau} \times L_{h\tau}$ of the same cycle are separated by a comma. If, for a component h of a cycle τ, the relative duration is one single time unit (i.e. $L_{h\tau} = 1$), the term $K_{h\tau} \times L_{h\tau}$ is substituted with $K_{h\tau}$. If a component has zero consumption, the corresponding term is denoted explicitly by $0 \times L_{h\tau}$ (or 0, if the duration of the relative sub-period is one single time unit). An example of the notation adopted is reported here below.

Cycle type τ: $K_{1\tau}, K_{2\tau} \times L_{2\tau}, 0, K_{4\tau} \times L_{4\tau}, K_{5\tau} \times L_{5\tau}$.

This reads, for cycle type τ, as follows:

the cycle starts with one sub-interval (i.e. one time unit) with a (constant) power consumption of $K_{1\tau}$ *units (component/sub-period 1);*
$L_{2\tau}$ *sub-intervals (i.e.* $L_{2\tau} > 1$ *time units) follow, with a (constant) power consumption of* $K_{2\tau}$ *units (component/sub-period 2);*
one sub-interval (i.e. one time unit) follows with a (constant) zero power consumption (component/sub-period 3);
$L_{4\tau}$ *sub-intervals(i.e.* $L_{4\tau} > 1$ *time units) follow with a (constant) power consumption of* $K_{4\tau}$ *units (component/sub-period 4);*
$L_{5\tau}$ *sub-intervals(i.e.* $L_{5\tau} > 1$ *time units) follow with a (constant) power consumption of* $K_{5\tau}$ *units*

(component/sub-period 5);
the cycle total duration is $1 + L_{2\tau} + 1 + L_{4\tau} + L_{5\tau}$ *time units.*

The fifty types of cycles reported in Table 1 are utilized in test sets A, B, C, D, E and G. In test set G, where three generic resources are considered, the electrical power is replaced by the first (generic) resource. In this case, the consumptions (per cycle type) appearing in Table 1 are interpreted in terms of the first generic resource units. This table also reports the maximum number of cycles admissible and these limits shall hold for all test sets from A to G (including F).

Cycle type	Power consumption (units)	Max. No. of cycles	Cycle type	Power consumption (units)	Max. No. of cycles
1	1,2,1	700	26	2x23	100
2	2,1,1	700	27	2x10,3x13	50
3	1,5,3	300	28	1,2x9,7x3,1x12	50
4	1x2,5,7,1	500	29	2x27	100
5	1,2,5x2,2	200	30	1x25,5x3,1x2	70
6	1,3,4,7,3	150	31	2x30	50
7	2,3x4,4x2	150	32	1x30	100
8	2,3x2,4x2,6,5	100	33	2x10,1x21	70
9	1,3,5x2,7x2,5	100	34	2x11,0,1x19	70
10	1x2,2,3,7,9x2,1x3	100	35	1x30,2x2	100
11	2x3,3x8	100	36	3x3,1x30	70
12	3x3,4x8	100	37	1x30,3x3	70
13	5x5,11x6	30	38	1x25,2x5,3x3	70
14	3x5,11x6,2x4	30	39	1x30,7,1x3	70
15	4x5,13x6,2x4	30	40	1x31,5x4	50
16	1x5,15x3,2x9	50	41	2,1x34	70
17	1x5,15x5,2x7	30	42	5,1x34	70
18	1x5,17x12	30	43	1x33,5x2	70
19	1x10,5x5,1x4	70	44	2x35,3	50
20	1x7,5x7,1x5	70	45	3x15,2x15,1x6	30
21	1x9,5x7,1x3	70	46	2x30,1x7	50
22	1x9,13x7,1x3	30	47	2x30,3x7	30
23	1x3,21x3,1x15	50	48	2x30,15,3x6	30
24	1x7,21x3,1x11	50	49	1x20,3x18	50
25	1x13,25x3,1x5	30	50	3x9,1x30	50

Table 1 Basic cycle characterization by their duration / power consumption

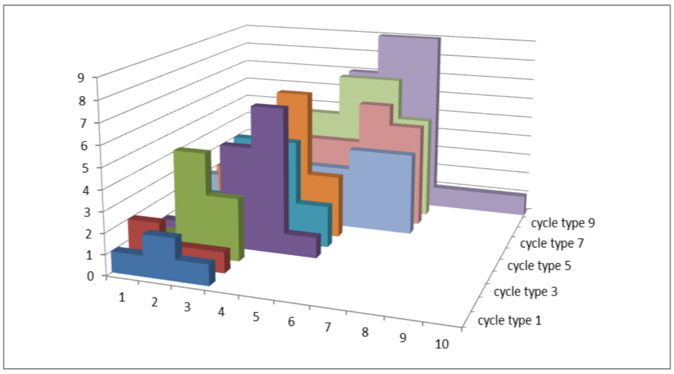

Fig. 2 Graphical representation of cycle types 1 to 10

Fig. 3 Graphical representation of cycle types 11 to 20

Fig. 4 Graphical representation cycle types 21 to 30

Fig. 5 Graphical representation of cycle types 31 to 40

Fig. 6 Graphical representation of cycle types 41 to 50

Figs. 2,3,4,5 and 6 provide a graphical representation of the cycle types considered. Each figure includes (in sequence) ten cycle types (some have been shifted to the right, in order to make the picture clearer).

The Basic test set consists of 25 instances, corresponding to a total time elapse of 100 units (i.e. 100 sub-intervals). Table 2 reports their sequential number in the first column. The second column indicates, for each test, the cycle types (from Table 1) that are available. The third column of the table shows the minimum number of cycles requested for each type. The last two columns report the results obtained, in terms of solution quality and computational effort.

Test	Cycle type	Min. No. of cycles	Energy exploitation (%)	CPU time (sec)
1	1-10	1-10: >0	95.6	298
2	11-20	1-20: >0	84.27	8
3	1-10; 21-30	1-10: > 0; 21-30: >0	95.12	299
4	1-10; 31-40	1-10: > 0; 31-40: > 0	96.03	13*
5	1-10; 41-50	1-10: > 0; 41-50: >0	96.39	4*
6	1-30	1-30: >0	95.48	269
7	1-10; 21-40	1-10: >0; 21-40: >0	95.08	283
8	1-10; 31-50	1-10: >0; 31-50: >0	96.23	44*
9	11-40	11-40: >0	92.0	190
10	11-20; 31-50	11-20: >0; 31-50: >0	96.39	296
11	1-20; 31-40	1-20: >0; 31-40: >0	96.95	117
12	11-30; 41-50	11-30: >0; 41-50: >0	93.42	299
13	11-30; 41-50	41-50: >0	94.09	153

14	1-40	11-20: >0	97.26	114*
15	1-40	11-30: >0	95.72	180*
16	1-40	11-40: >0	96.11	292
17	11-50	11-20: >0	96.31	276
18	11-50	11-30: >0	91.2	143
19	11-50	11-40: >0	89.34	141
20	11-50	21-50: >0	91.59	299
21	1-50	11-20: >0	95.28	190
22	1-50	11-30: >0	94.81	144
23	1-50	11-40: >0	95.56	225
24	1-50	21-50: >0	95.4	287
25	1-50	1-10: >6; 11-20: >0	95.72	299

Table 2 Basic test set instances and performance results

3.2 Test sets A, B, C, D, E and F

Test set A, similarly to the Basic one, considers a constant power capacity of 25 units. It consists of subsets A1, A2, A3 and A4. Subset A1 coincides with the Basic test set. Subsets A2, A3 and A4 differ from the Basic set only for the total time availability. The following time periods have been considered:

- A1: [0,100] time units
- A2: [0,150] time units
- A3: [0,200] time units
- A4: [0,250] time units

Similarly, subsets B1, B2, B3, B4, … , F1, F2, F3 and F4 are defined over the same time periods (i.e. [0,100], [0,150], [0,200] and [0,250] time units). Tests B, C, D and E are derived from test set A by changing the power capacity only. Four different power functions, not constant any longer, were hence introduced. They are represented in Figs. 7, 8, 9 and 10 respectively (and reported in detail in the Appendix).

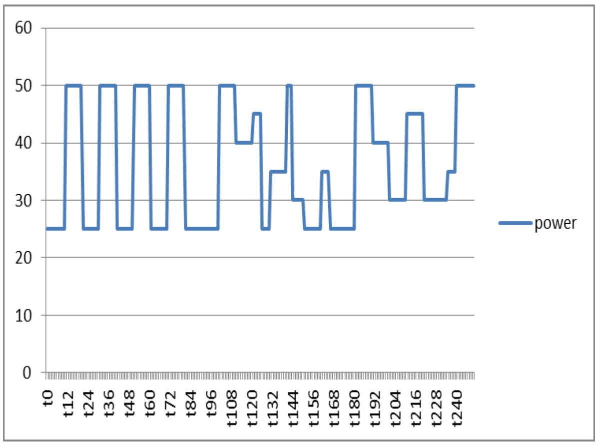

Fig. 7 Test set B power function

Fig. 8 Test set C power function

Fig. 9 Test set D power function

Fig. 10 Test set E power function

For test sets A, B, C, D and E, the constants W_v (associated with the power step-functions, see (5)) are integer, as well as $K_{h\tau}$ (representing, for each component h of each cycle τ, the power consumption, see Section 3.1). Test set F was purposely introduced to consider the case where both W_v and $K_{h\tau}$ may take, instead, any (non-negative) real values. Test set F was obtained from test set B by adding/subtracting fractional quantities, between 0 and 1, to/from the values corresponding both to the power function and the power consumption (see the Appendix for more details). It is understood that for all test sets from A to F, all data relevant to the cycles (as reported in Tables 1) are considered, as well as the additional conditions introduced in Section 3.1.

The relative computational results, in terms of energy exploitation percentage and CPU time (seconds) are reported in Table 3. There, each test set (i.e. A, B, C, D, E and F) is partitioned into the corresponding subsets of 25 tests each (i.e. A1, A2, A3, A4, … , F1, F2, F3 and F4). For each subset, the average of the energy exploitation percentage and CPU time (seconds) is reported.

Test Subset	Energy exploitation (%) average	CPU time (sec)
A1	94.45	195
A2	94.18	245
A3	94.17	233
A4	92.4	221
B1	94.81	225
B2	94.28	253
B3	93.05	242
B4	90.19	159
C1	94.3	210
C2	94.2	254
C3	93.04	216
C4	90.71	190

Test Subset	Energy exploitation (%) average	CPU time (sec)
D1	93.98	204
D2	93.87	259
D3	93.07	196
D4	91.7	219
E1	93.33	245
E2	91.99	254
E3	91.51	194
E4	89.63	217
F1	92.07	201
F2	91.31	220
F3	88.94	186
F4	87.73	168

Table 3 Performance results of test sets A, B, C, D, E and F

Table 4 shows, for test subsets A4, B4, C4, D4, E4 and F4, the MILP model matrix dimension, in terms of number of rows, non-zero elements and 0-1 variables after the (MIP) pre-processing carried out by the solver.

Test A4-F4	No. of rows	No. of (0-1) variables	No. of non-zero elements
1	2664	2450	29010
2	2471	2342	72581
3	4741	4733	130388
4	4556	4643	161840
5	4470	4600	174796
6	6968	7094	202078
7	7004	6765	379560
8	6356	6788	307563

9	6433	6816	305470
10	6162	6691	350098
11	6782	7001	233448
12	6328	6784	314225
13	6331	6784	315448
14	8824	9282	327921
15	8834	9282	330202
16	8844	9282	332390
17	8204	8972	444881
18	8214	8972	447162
19	8224	8972	449350
20	8225	8972	449369
21	11334	11321	656327
22	10624	11427	473780
23	10634	11427	475968
24	10637	11427	477183
25	10624	11427	473954

Table 4 Test subsets A4 to F4 matrix size/density

3.3 Test set G

This group of tests contains instances with three different resources each. It was derived from the previous, by substituting the power function with a generic one and adding resources 2 and 3. The power consumption values corresponding to each cycle type (as reported in Table 1) were kept unaltered and associated with resource 1 (no longer necessarily representing the power), 2 and 3 (the same cycle profiles were in fact assumed for the three resources). Three new functions were defined for resource 1, 2 and 3, respectively. These are represented in Fig. 11 (relevant details are reported in the Appendix).

As is gathered, when more than one resource is involved, their total exploitation is generally expected to decrease markedly. The MILP model matrix, on the other hand, increases significantly. The overall performance results obtained for subsets G1, G2, G3 and G4 (corresponding, as in the previous cases, to the time periods [0,100], [0,150], [0,200] and [0,250] time units) are summarized in Table 5.

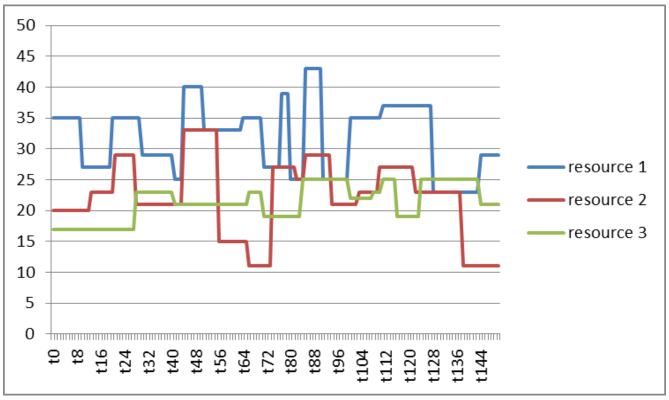

Fig. 11 Test G set resource 1-2-3 functions

3.4 Tests extracted from Fekete's and Shepers' set

The case studies reported in this Section refer to the sets '2D constrained non-guillotine NGCUTFS, file ngcutfs1, provided by Fekete and Shepers (see www.or.deis.unibo.it/research_pages/ORinstances/ORinstances; www.fe.up.pt/esicup) that are expressed in terms of classical two-dimensional knapsack problems, without rotations.

4 Conclusions

This work is inspired by very challenging issues arising in space logistics, where, quite often, the activity requested has to be carried out in extremely limited conditions, both in terms of time and resource capacity. The necessity of optimizing the scheduling of activities, subject to a number of tight constraints, is nonetheless becoming, day after day, ever more demanding in several contexts apart from space.

The problem tackled in this chapter, addresses the cases where the resource capacities, in a given time elapse, are not constant. The activities themselves are characterized by non-constant resource request profiles. The case of a single resource, identified with electrical power, is discussed firstly, pointing out the relevant modelling aspects. An MILP formulation, based on a time-indexed approximation approach is provided. Extensions of the basic model to multiple-resource scenarios are discussed, as well as the introduction of additional conditions. Hints on possible applications of the methodology adopted are put forward and an in-depth experimental analysis is provided. The investigation of ad-hoc computational strategies to solve the relevant models ever more efficiently might represent the objective of future research.

5 References

1. Agnetis, A., Billaut, J.C., Gawiejnowicz, S., Pacciarelli, D., Soukhal, A.: Multiagent Scheduling. Springer-Verlag Berlin, Heidelberg (2014)
2. van den Akker, J.M., Hurkens, C.A.J., Savelsbergh, M.W.P.: Time-Indexed Formulations for Machine Scheduling Problems: Column Generation. INFORMS Journal on Computing, **12**(2), 111-124 (2000)
3. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener. Comput. Syst. **28**(5), 755-768 (2012)
4. Bidot, J., Vidal, T., Laborie, P., Beck, J. C.: A theoretic and practical framework for scheduling in a stochastic environment. Journal of Scheduling, **12(**3), 315-344 (2009)
5. Błażewicz, J., Ecker, K.H., Pesch, E., Schmidt, G, Weglarz, J.: Handbook on Scheduling, International Handbooks on Information Systems. Springer-Verlag Berlin, Heidelberg (2007)
6. Brucker, P., Knust, S.: Complex Scheduling. Springer-Verlag Berlin, Heidelberg (2012)
7. Pedro M., Castro , Oliveira J.F.: Scheduling inspired models for two-dimensional packing problems. European Journal of Operational Research **215**, 45-56 (2011)
8. Chen Z., Chyu C.: An Evolutionary Algorithm with Multi–Local Search for the Resource-Constrained Project Scheduling Problem. Intelligent Information Management **2**, 220-226 (2010)
9. Coelho, J., Vanhoucke, M.: Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers European Journal of Operational Research **213**(1), 73-82 (2011)
10. Damak, N., Jarboui, B., Siarry, P., Loukil, T.: Differential evolution for Solving multi-mode resource-constrained project scheduling problems. Computers and Operations Research **36**(9), 2653-2659 (2009)
11. Duy Ha, L., Ploix, S. , Zamai, E., Jacomino, M.: Control of energy consumption in home automation by ressource constraint scheduling. The 15th International Conference on Control System and ComputerScience, Bucharest, Romania, May 25-27 2005
12. Fasano, G.: Solving Non-standard Packing Problems by Global Optimization and Heuristics. SpringerBriefs in Optimization**,** Springer Science + Business Media, New York (2014)
13. Gonzalez, F., Ramies, R.D.: Multi-objective optimization of the resource constrained project scheduling problem (RCPSP) a heuristic approach based on the mathematical model. Int. J. Comput. Sci. Appl. **2**(2),1-13 (2013)
14. Hartmann, S.: Packing Problems and Project Scheduling Models: an Integrating Perspective. Journal of Operational Research Society **51**, 1083-1092 (2000)
15. Hartmann, S.: Project scheduling under limited resources: Models, methods, and applications. Lecture Notes in Economics and Mathematical Systems, **478**, Berlin: Springer (2013)
16. IBM Corporation: ILOG CPLEX Optimizer. High performance mathematical optimization engines. IBM Corporation Software Group, NY 10589 U.S.A. WSD14044-USEN-01 (2010)
17. Jaberi, M.: A multi-objective resource-constrained project-scheduling problem using mean field annealing neural networks. J. Math. Comput. Sci. **9**, 228-239 (2014)
18. Kabra, S., Shaik, M.A., Rathore, A. S.: Multi-period Scheduling of a Multistage Multiproduct Bio-Pharmaceutical Process. Computers & Chemical Engineering, 52, 95-103 (2013)
19. Lee, Y. C., Zomaya, A. Y.: Rescheduling for reliable job completion with the support of clouds. Future Generation Computer Systems **26**, 1192-1199 (2010)

20. Li, J., Misener, R., Floudas, C.A.: Continuous-time Modeling and global optimization approach for scheduling of crude oil operations. AIChE Journal, **58**(1), 205-226 (2012)
21. Liu, Y., Dong, H., Lohse, N., Petrovic, S., Gindy, N.: An investigation into minimising total energy consumption and total weighted tardiness in job shops. Journal of Cleaner Production, **65**, 87-96 (2014)
22. Malakooti, B.: Operations and Production Systems with Multiple *Objectives*. John Wiley & Sons. ISBN 978-1-118-58537-5 (2013)
23. Mendes J.J.M., Gonvalces J.F., Resende M.G.C.: A random key based genetic algorithm for the resource constrained project scheduling problem. Computers & Operations Research **36**, 92-109 (2009)
24. Miyamoto, T., Mori, K., Izui, Y., Kitamura, S.: A Study of resource constraint project scheduling problem for energy saving DOI: 10.1109/ICSSE.2014.6887897 Conference: International Conference on System Science and Engineering,ICSSE (2014)
25. Peteghem, V.V., Vanhoucke, M.: A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. European Journal of Operational Research, **201** (2), 409-418 (2010)
26. Pinedo, M.L.: Planning and Scheduling in Manufacturing and Services, Springer, New York (2005)
27. Pisinger, D., Sigurd, M.: The two-dimensional bin packing problem with variable bin sizes and costs. Discrete Optimization **2**(2), 154-167 (2005)
28. Shaik, M.A., Floudas, C.A.: Novel Unified Modeling Approach for Short-term Scheduling. Industrial & Engineering Chemistry Research, 48, 2947-2964 (2009)
29. Xu, J.P., Zeng, Z.Q., Han, B., Lei, X.: A dynamic programming-based particle swarm optimization algorithm for an inventory management problem under uncertainty. Engineering Optimization, **45**(7), 851-880 (2013)
30. Yannibelli, V., Amandi, A.: Hybridizing a multi-objective simulated annealing algorithm with a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem. Expert Syst. Appl., **40**, 2421-2434 (2013)
31. Zhang, Z., Chen, J.: Solving the spatial scheduling problem: a two-stage approach. International Journal of Production Research **50**(10),2732-2743 (2012)
32. Ziarati, K., Akbari, R., Zeighami, V.: On the performance of bee algorithms for resource-constrained project scheduling problem. Applied Soft Computing Journal, **11**(4), 3720-3733 (2011)