The Electromagnetism Meta-heuristic Applied to the Resource-Constrained Project Scheduling Problem

Dieter Debels[1] and Mario Vanhoucke[1,2]

[1] Ghent University, Faculty of Economics and Business Administration, Hoveniersberg 24, Ghent 9000, Belgium
[2] Vlerick Leuven Gent Management School, Operations & Technology Management Centre, Reep 1, Ghent 9000, Belgium
{dieter.debels, mario.vanhoucke}@ugent.be

Abstract. Recently, an electromagnetism (EM) heuristic has been introduced by Birbil and Fang (2003) to solve unconstrained optimization problems. In this paper, we extend the EM methodology to combinatorial optimization problems and illustrate its effectiveness on the well-known resource-constrained project scheduling problem (RCPSP). We present computational experiments on a standard benchmark dataset, compare the results of the different modifications on the original EM framework with current state-of-the-art heuristics, and show that the procedure is capable of producing consistently good results for challenging instances of the problem under study. We also give directions for future research in order to further explore the potential of this new technique.

1 Introduction

The problem under study is the well-known resource-constrained project scheduling problem (RCPSP). The RCPSP can be stated as follows. A set of activities N, numbered from 1 to n ($|N| = n$), is to be scheduled without pre-emption on a set R of renewable resource types. Activity i has a deterministic duration $d_i \in$ IN and requires $r_{ik} \in$ IN units of resource type k, $k \in R$, which has a constant availability a_k throughout the project horizon. We assume that $r_{ik} \leq a_k$ for $i \in N$ and $k \in R$. The dummy start and end activities 1 and n have zero duration while the other activities have a non-zero duration. The dummies also have zero resource usage. A is the set of pairs of activities for which a finish-start precedence relationship with time lag 0 exists. We use S_i (P_i) to denote the set of immediate successors (predecessors) of activity i and S'_i (P'_i) to refer to the set of all (immediate and transitive) successors (predecessors) of this activity. We assume graph G(N,A) to be acyclic. A schedule S is defined by an n-vector of start times $s = (s_1, ..., s_n)$ which implies an n-vector of finish times e ($e_i = s_i + d_i, \forall\, i \in N$). A schedule is said to be feasible if the precedence and resource constraints are satisfied. The objective of the RCPSP is to find a feasible schedule such that

E. Talbi et al. (Eds.): EA 2005, LNCS 3871, pp. 259–270, 2006.

the schedule makespan e_n is minimized. In this paper we report results for the application of a recent heuristic technique, electromagnetism (EM) [4]. EM is an 'evolutionary' algorithm that was originally developed for the optimization of unconstrained continuous functions. As we modify the technique to solve the RCPSP, we show that EM can also be used for combinatorial problems.

2 The Electromagnetism Meta-heuristic

Birbil and Fang [4] propose a so-called electromagnetism optimization heuristic for unconstrained optimization problems, i.e. the minimization of non-linear functions. These optimization problems with bounded variables can be modeled as depicted at the left side of Fig. 1. At the right side, we show how the RCPSP can be reformulated as an unconstrained optimization problem. To obtain a Euclidean solution space, we opt for a schedule representation in random-key (RK) format [17]. To transform an RK vector $x \in \mathrm{IR}^n$ into a schedule $S = \sigma(x)$ with an associated makespan $e_n(\sigma(x))$, a schedule generation scheme (SGS) is necessary. We make use of the serial SGS, as it is sometimes impossible to reach an optimal solution with the parallel SGS [14]. In the remainder of the paper we assume that a higher RK value corresponds to a lower priority of the activity. By setting $l_i = 0$ and $u_i = n$, we assume that each priority element of an RK vector is a real value between 1 and n. In order to decrease the solution space of the search process, we adapt this vector with new lower and upper values to $l_i = |P'_i| + 1$ and $u_i = n - |S'_i|$. In doing so we increase the likelihood that the obtained solution corresponds to a precedence-feasible priority structure where each activity has a lower priority (i.e. a higher RK value) than its predecessors.

Consider the example project presented in Fig. 2. This project network contains 9 non-dummy activities for which the duration is given above the node and the resource requirement for the single resource below the node. The corresponding lower and upper values (l_i and u_i) for the RK value of each activity i are given between brackets. Assume that the resource availability equals 2, then Fig. 3 depicts two feasible schedules for the example project.

The EM heuristic assumes a multidimensional solution space where each point x represents a solution. A charge is associated to each point, related to

General	RCPSP
minimize $f(x)$	minimize $e_n(\sigma(x))$
s.t.	s.t.
$\boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]$	$\boldsymbol{x} \in [\boldsymbol{l}, \boldsymbol{u}]$
where $[\boldsymbol{l}, \boldsymbol{u}] = \{\boldsymbol{x} \in \mathrm{IR}^D \mid l_i \leq x_i \leq u_i,\ i = 1, \ldots, D)\}$	where $[\boldsymbol{l}, \boldsymbol{u}] = \{\boldsymbol{x} \in \mathrm{IR}^n \mid l_i \leq x_i \leq u_i,\ i = 1, \ldots, n)\}$
with D the dimension of the solution space	with n the number of activities

Fig. 1. Formulation of unconstrained optimization problems

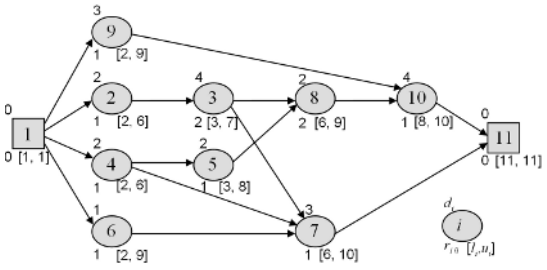

Fig. 2. Example project network with 9 non-dummy activities

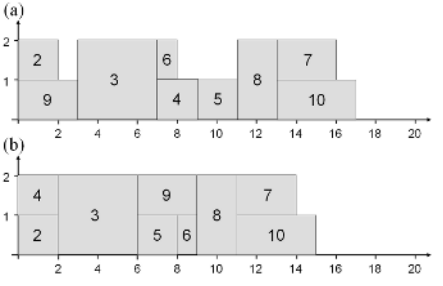

Fig. 3. Two feasible schedules for the example project

the objective function value $f(x)$ associated with the solution point x. As in evolutionary algorithms, a population is created in which each solution point will exert attraction or repulsion on other points, of which the magnitude is proportional to the product of the charges and inversely proportional to the distance between the points (Coulomb's Law). The principle behind the algorithm is that inferior solution points will prevent a move in their direction by repelling other solution points in the population, and that attractive points will facilitate moves in their direction. The generic pseudo-code for the EM algorithm is as follows:

```
Algorithm EM
    iter :=1
    while stop condition not satisfied do
        compute_forces
        apply_forces
        local_search
        iter++
    endwhile
```

The function EM contains three subroutines (*compute_forces*, *apply_forces* and *local_search*), that are iteratively applied as long as the stop condition is not satisfied. The total force exerted on each point by all other points is calculated in the function *compute_forces* and depends on the charge of the point under consideration as well as on the points exerting the force. The charge

of the k^{th} population point x^k is determined by its objective function value $f(x^k)$ in relation to the objective function value $f(x^{best})$ of the current best point in the population. For a minimization problem, the charge q^k of point x^k is determined according to eq. 1. In the first term we calculate the value of q^k as given by Birbil and Fang [4], and in the second term we translated the formulation to the RCPSP context. Note that m represents the population size.

$$q^k = e^{\left(\frac{-D.\left(f\left(x^k\right)-f\left(x^{best}\right)\right)}{\sum_{l=1}^{m}\left(f(x^l)-f(x^{best})\right)}\right)} = e^{\left(\frac{-n.\left(e_n\left(\sigma((x^k)\right)-e_n\left(\sigma((x^{best})\right)\right)}{\sum_{l=1}^{m}\left(e_n(\sigma((x^l))-e_n(\sigma((x^{best}))\right)}\right)} \tag{1}$$

The total force exerted on a point by all other points is calculated in a similar way with Coulomb's law and is inversely proportional to the distance between the points and directly proportional to the product of their charges. The set of force vectors $F^k(k = 1, \ldots, m)$ exerted on the corresponding point x^k is determined as shown in eq. 2. The point with a relatively good objective function value attracts the other one, the point with the inferior objective function value repels the other. In $\|\text{dist}(x^l, x^k)\|$ we measure the normalized distance between two points (schedules) x^k and x^l. The distance equals the sum of the absolute deviations of the priority values x^k and x^l of each activity i, i.e. $\text{dist}(x^l, x^k) = \sum_{i=1}^{n} |x_i^l - x_i^k|$. In order to normalize the distance measure to the interval [0, 1], we set $\|\text{dist}(x^l, x^k)\| = \text{dist}(x^l, x^k)/\text{dist}_{max}$ with dist_{max} the maximum of all distances between each pair of points, i.e. $\text{dist}_{max} = \max_{l=1,\ldots,m;k=1,\ldots,m} \text{dist}(x^l, x^k)$. Thus, points with a better objective function value attract point x^k, while points with an inferior objective function value repel x^k.

$$F^k = \sum_{l=1,l\neq k}^{m} \left\{ \begin{array}{l} (x^l - x^k)\left(\frac{q^l.q^k}{\|\text{dist}(x^l,x^k)\|}\right) \text{ if} f(x^l) < f(x^k) \\ (x^k - x^l)\left(\frac{q^l.q^k}{\|\text{dist}(x^l,x^k)\|}\right) \text{ if} f(x^l) \geq f(x^k) \end{array} \right\} \tag{2}$$

The movement according to the resulting forces is performed in $apply\_forces$ and is shown in eq. 3. The move is based on the normalized force vector $\|F^k\| = F^k / \max_{i=1,\ldots,n}(F_i^k)$. Thus, the original force vector F^k only identifies the direction of the move. The magnitude of each move is determined by a randomly selected parameter λ, generated from a uniform distribution from the interval [0,1] and also by the lower value l_i and upper value u_i for the priority value x_i^k belonging to the i^{th} activity of population element k.

$$x_i^k = \left\{ \begin{array}{l} x_i^k + \lambda\|F_i^k\|(u_i - x_i^k) \text{ if} F_i^k > 0 \\ x_i^k + \lambda\|F_i^k\|(x_i^k - l_i) \text{ if} F_i^k \leq 0 \end{array} \right\} \tag{3}$$

After the application of the forces on the population elements, $local\_search$ aims to improve the newly obtained solution points. In the original version [4], a local search technique that explores the immediate (Euclidian) neighbourhood of individual points is proposed. However, for the RCPSP it is beneficial to use the iterative forward/backward scheduling technique [21] as a simple and effective local search technique. To obtain an improved schedule, the technique iteratively performs backward and forward passes. A backward pass transforms

a left-justified schedule in a right-justified schedule by scheduling the activities backwards in decreasing order of their finish times. A forward pass transforms a right-justified schedule in a left-justified schedule by scheduling the activities forwards in increasing order of their start times. In doing so, the schedule makespan of each intermediate schedule is never higher than the makespan of the previous one.

To the best of our knowledge, the EM philosophy has only been used for scheduling projects by [7]. However, these authors present a *scatter search* algorithm for the RCPSP, and seed their algorithm with very basic principles taken from the electromagnetism philosophy. More precisely, they restrict the use of the EM philosophy to the description of the hybrid two-point/electromagnetism crossover operator. However, a closer look to this hybrid crossover reveals that

- Forces are only calculated based on *one* other population-element. This is not in line with the basic EM philosophy in which a point exerts a force on *all* other points.
- The forces are not related to the distance between solutions. This is in contradiction to the EM philosophy in which the magnitude of the force is inversely proportional to the distance between points, in order to follow the law of Coulomb.

In section 3 of the current paper, we present a step-wise adaptation of our EM algorithm to cope with the RCPSP, following the framework as proposed by [4].

3 Computational Experiments

We have coded the procedure in Visual C++ 6.0 and performed computational tests on an Acer Travelmate 634LC with a Pentium IV 1.8 GHz processor using the well-known PSPLIB dataset [19]. This set contains the subdatasets J30, J60, J90 and J120 with problem instances of 30, 60, 90 and 120 activities. In section 3.1, we describe a step-wise adaptation of the algorithm of section 2 in order to improve the performance. In doing so, we rely on specific characteristics of the RCPSP. Section 3.2 compares the performance of our EM heuristic with other state-of-the-art results.

3.1 Using Problem-Specific Characteristics of the RCPSP

Table 1 reports the results for our step-wise improvement scenarios as discussed in the following subsections, based on a run with 5,000 schedules. The column labelled "Basic" reports the results for the basic EM meta-heuristic of section 2. The following columns report the results for the different modifications on this basic EM algorithm. More details are given in the following subsections. The rows labelled "Avg.Dev.Lb" report the average deviation from the optimal solution (J30 instances) or from the critical path based lower-bound (J60, J90 and J120 instances). The rows labelled "Avg.CPU" indicate the average computation time to solve a problem instance (in seconds). For each adaptation, we have finetuned

Table 1. Results for the step-wise improvement scenarios for the basic EM meta-heuristic

Dataset	Basic	SRK	MUT	NBH	SUB
Avg.Dev.Lb					
J30	0.20%	0.22%	0.10%	0.11%	0.12%
J60	11.94%	11.50%	11.47%	11.43%	11.29%
J90	11.81%	11.06%	11.06%	11.05%	10.89%
J120	36.50%	34.32%	34.32%	34.21%	33.98%
Avg.CPU					
J30	0.09s	0.07s	0.08s	0.07s	0.06s
J60	0.16s	0.15s	0.15s	0.14s	0.13s
J90	0.26s	0.23s	0.23s	0.20s	0.19s
J120	0.36s	0.36s	0.36s	0.31s	0.34s
m					
J30	40	35	30	30	25
J60	30	40	25	30	25
J90	40	35	35	30	25
J120	40	35	35	30	15
p_{mut}					
J30	-	-	0.10	0.10	0.10
J60	-	-	0.01	0.01	0.01
J90	-	-	0.00	0.00	0.00
J120	-	-	0.00	0.00	0.00

the algorithm by setting the population size m to an optimal value. These values are given in the rows "m". As an example, the table reveals that the basic version of the algorithm reports the best results for a population size of 40 for the J30, J90 and J120 instances and a population size of 30 for the J60 instances. The rows labelled "p_{MUT}" are used to display the percentage of mutation, which will be discussed in section 3.1.2.

3.1.1 Topological-Order Representation or Standardized RK (SRK)

In the RK representation, each solution corresponds to a point in the Euclidian n-space, so that geometric operations can be performed on its components. Since this is one of the cornerstones of the EM method, we have adopted the RK representation in our EM-heuristic. However, the RK representation suffers from the fact that one schedule can have an infinite number of schedule representations. To deal with this problem, we propose to use a topological-order (TO) representation of schedules [30,31]. A TO representation in RK format of a schedule is any RK vector x for which $s_i < s_j$ implies $x_i < x_j$. To incorporate the TO condition, we change the RK representation to the so-called standardized RK (SRK) as suggested by [7]. More precisely, we first rank the activities according to their start times in the schedule, and then replace their priority values by the place in the ranking. In doing so, the SRK vector fits very well into the EM framework, since each vector element will have a value between l_i and u_i. As an example, the SRK vector for the schedule of Fig. 3(a) is $\{1, 2, 4, 5, 7, 5, 9, 8, 2, 9, 11\}$. Note that the SRK-value for each activity i always lies between $l_i = |P'_i| + 1$ and $u_i = n - |S'_i|$ and that we can only transform an RK vector x into SRK format $\pi(x)$ after the schedule generation. The results

Table 2. Input data for the example: the start RK vector and the two forces

Activity i	1	2	3	4	5	6	7	8	9	10	11
x_i^0	1	2	4	5	7	5	9	8	2	9	11
$\lambda^1 \cdot \|F_i^1\|$	0	0.26	-0.21	-0.09	-0.15	0.03	0.57	-0.05	0.14	-0.17	0
$\lambda^2 \cdot \|F_i^2\|$	0	-0.51	-0.37	-0.19	-0.44	0.02	-0.08	-0.18	0.09	0.57	0

of the incorporation of the TO representation by using the SRK can be seen in table 1 by comparing the columns labelled "Basic" and "SRK". The results show a beneficial effect for the J60, J90 and J120 sets and a negative effect for the J30 instances.

Although the SRK representation embeds the logic that early scheduled activities have a high priority, it also has a major drawback. The execution of a force on an SRK vector $x^0 = \pi(x^0)$ modifies the priority structure of the vector to an RK-vector x^1 which will be transformed by means of the serial SGS and the local search method to a schedule with a corresponding SRK notation $\pi(x^1)$. It is, however, possible that the resulting schedule (and consequently, the resulting SRK notation) is not different from the original one, i.e. $\pi(x^0) = \pi(x^1)$ while $x^0 \neq x^1$. This effect might prevent to exploit the advantages of the basic philosophy of the EM approach, which focuses on a gradual shift to other regions of the solution space. Due to the transformation from x^1 to $\pi(x^1)$, this gradual shift from x^0 to x^1 will be cancelled out, having an effect on the next moves of the meta-heuristic. More precisely, our tests revealed that 79% of the moves, performed on solutions in SRK format, result in a schedule for which $\pi(x^0) = \pi(x^1)$ for the 30-activity networks. This value decreases to 65% for the J60 instances, 56% for the J90 instances and only 14% for the J120 instances. Thus, the cancel-out problem is particularly relevant for small problem instances, as the solution space is too small to escape from a solution point.

Consider the example project of Fig. 2 and the two corresponding schedules of Fig. 3. The start vector x^0 corresponds to the schedule of Fig. 3(a). $\lambda^1\|F^1\|$ and $\lambda^2\|F^2\|$ are used to calculate the first and second move and are given in table 2. Table 3 displays the calculations of two moves in a sequence, based on the RK vectors while table 4 displays the calculations of these moves based on the SRK. In the first move of table 3, from x^0 to x^1, the RK-representation changes, but the corresponding schedule remains unchanged and is equal to the schedule of

Table 3. Illustration of the execution of two moves based on a RK vector

activity i	1	2	3	4	5	6	7	8	9	10	11
x_i^0	1	2	4	5	7	5	9	8	2	9	11
$\Delta x_i^0 = \lambda^1 \cdot \|F_i^1\| \cdot (u_i - x_i^0)$		1.03				0.11	0.57		0.96		
$\Delta x_i^0 = \lambda^1 \cdot \|F_i^1\| \cdot (x_i^0 - l_i)$	0		-0.21	-0.27	-0.62			-0.3		-0.17	0
$x_i^1 = x_i^0 + \Delta x_i^0$	1	3.03	3.79	4.73	6.38	5.11	9.57	7.7	2.96	8.83	11
$\Delta x_i^1 = \lambda^2 \cdot \|F_i^2\| \cdot (u_i - x_i^1)$						0.09			0.54	0.67	
$\Delta x_i^1 = \lambda^2 \cdot \|F_i^2\| \cdot (x_i^1 - l_i)$	0	-0.53	-0.29	-0.53	-1.48		-0.37	-0.3			0
$x_i^2 = x_i^1 + \Delta x_i^1$	1	2.5	3.5	4.2	4.9	5.2	9.2	7.4	3.5	9.5	11

Table 4. Illustration of the execution of two moves based on a SRK vector

activity i	1	2	3	4	5	6	7	8	9	10	11
x_i^0	1	2	4	5	7	5	9	8	2	9	11
$\Delta x_i^0 = \lambda^1 \lVert F_i^1 \rVert (u_i - x_i^0)$		1.03				0.11	0.57		0.96		
$\Delta x_i^0 = \lambda^1 \lVert F_i^1 \rVert (x_i^0 - l_i)$	0		-0.21	-0.27	-0.62			-0.3		-0.17	0
$x_i^1 = x_i^0 + \Delta x_i^0$	1	3.03	3.79	4.73	6.38	5.11	9.57	7.7	2.96	8.83	11
$x_i^1 = \pi(x_i^1)$	1	2	4	5	7	5	9	8	2	9	11
$\Delta x_i^1 = \lambda^2 \lVert F_i^2 \rVert (u_i - x_i^1)$		0				0.09			0.63	0.57	
$\Delta x_i^1 = \lambda^2 \lVert F_i^2 \rVert (x_i^1 - l_i)$	0		-0.37	-0.58	-1.75		-0.32	-0.36			0
$x_i^2 = x_i^1 + \Delta x_i^1$	1	2	3.63	4.42	5.25	5.09	8.68	7.64	2.63	9.57	11
$x_i^2 = \pi(x_i^2)$	1	2	4	5	7	5	9	8	2	9	11

Fig. 3(a). After the second move, from x^1 to x^2, the RK-representation belongs to the schedule depicted in Fig. 3(b). Table 4, however, cannot escape from the schedule of Fig. 3(a) since the second move from $\pi(x^1)$ to $\pi(x^2)$ also results in the same schedule. This is due to the transformation of the RK vector x^1 to the SRK vector $\pi(x^1)$ after a makespan evaluation, which cancels out the gradual shift of the move from x^0 to x^1. In the next sub-section, we describe our mutation approach to overcome this cancel-out problem.

3.1.2 Diversification Using Mutation (MUT)

In order to prevent the population from becoming overly homogeneous, we introduce a basic version of diversification using mutation, by modifying randomly chosen priority values of the vector x to a value uniformly chosen between $l_i = |P'_i| + 1$ and $u_i = n - |S'_i|$. This mutation is imposed right after a force is executed, followed by a makespan evaluation.

In table 1, we use p_{MUT} to denote the percentage of activities that are subject to this mutation per move. The results reveal that mutation is only beneficial for the J30 and J60 instances. For the J30 instances, we modify 10% of the priority values per move, i.e. in each move, three activities receive a new priority value randomly generated from the interval $[l_i, u_i]$. The average deviation decreases from 0.22% to 0.10%. For the J60 instances, mutation is only beneficial to a small extent (from 11.50% to 11.47% deviation) with a mutation rate of only 1%. For the J90 and J120 instances, mutation has no beneficial effect. These results confirm that mutation can help to overcome the problem mentioned in the previous section. Since the use of the SRK notation could possibly cancel out the gradual shifts in the solution space, diversification using mutation will be necessary to escape from a particular schedule. This problem was particularly relevant for the J30 instances and - to a smaller extent - for the J60 instances.

3.1.3 Extended Neighbourhood (NBH)

In the original procedure [4] a parameter λ is generated from a uniform distribution between 0 and 1, i.e. $\lambda \in \text{U}(0, 1)$, in order to move from one schedule to

another (see eq. 3). We have extended this method by generating more schedules out of a schedule by generating more values for λ. To that purpose, we have divided the interval [0, 1] in equal parts and tested a number of scenarios with 2, 3, 4, 5 and 6 different values for λ and, consequently, up to 6 new schedules per move. Tests have revealed that improved results can be found for J60, J90 and J120 by generating two new schedules with the following parameter values: $\lambda^1 \in \mathrm{U}(0.2, 0.6)$ and $\lambda^2 \in \mathrm{U}(0.6, 1)$. Afterwards, the algorithm selects the best schedule to enter the population. Note that moves with $\lambda < 0.2$ are excluded since this often leads to the cancel-out effect described in section 3.1.1.

3.1.4 Exert the Force F on a Sub-schedule (SUB)

Based on the calculated forces and resulting attraction or repulsion, points are transformed, i.e. moved in the Euclidian space, resulting in a new population. During each move, forces are exerted on the priority value of each activity. We generalize this concept by allowing forces to act only in a particular subset of the dimensions or activities. We randomly select $p_{min} \in [1, n-1]$ and set $p_{max} = p_{min} + \tau$ with τ chosen randomly within $[1/4.e_n(\sigma(x)), 3/4.e_n(\sigma(x))]$. Then, we update only the RK values between p_{min} and p_{max} (inclusive) according to the forces exerted in these dimensions. Note that due to the SRK

Table 5. Comparative computational results

	J30		J60		J120	
Algorithm	Avg.Dev.Lb	Rank	Avg.Dev.Lb	Rank	Avg.Dev.Lb	Rank
Valls, Ballestin and Quintanilla [32]	0.06	2	11.10	1	32.54	1
Debels, De Reyck, Leus, Vanhoucke [7]	0.11	5	11.10	1	33.10	2
Valls, Ballestin and Quintanilla.[33]	0.20	12	11.27	5	33.24	3
Kochetov and Stolyar [13]	0.04	1	11.17	3	33.36	4
Alcaraz, Maroto and Ruiz [2]	0.06	2	11.19	4	33.91	5
Our procedure	**0.10**	**4**	**11.29**	**6**	**33.94**	**6**
Valls, Ballestin and Quintanilla.[33]	0.28	16	12.35	16	34.02	7
Tormos and Lova [29]	0.13	7	11.62	7	34.41	8
Hartmann [9]	0.22	13	11.70	8	35.39	9
Merkle, Middendorf and Schmeck [23]	-		-		35.43	10
Tormos and Lova [28]	0.17	10	11.82	9	35.56	11
Tormos and Lova [27]	0.16	8	11.87	11	35.81	12
Alcaraz and Maroto [1]	0.12	6	11.86	10	36.57	13
Hartmann [8]	0.56	23	13.32	24	36.74	14
Bouleimen and Lecocq [5]	0.23	14	11.90	13	37.68	15
Nonobe and Ibaraki [24]	0.16	8	12.18	15	37.88	16
Coelho and Tavares [6]	0.33	17	12.63	18	38.41	17
Hartmann [8]	0.25	15	11.89	12	38.49	18
Schirmer [25]	0.44	18	12.58	17	38.70	19
Kolisch [15]	0.53	21	13.23	22	38.75	20
Kolisch [15, 17]	1.28	26	13.21	21	38.77	21
Kolisch and Drexl [16]	0.52	20	13.06	20	40.45	22
Coelho and Tavares [6]	0.54	22	13.31	23	40.46	23
Leon and Ramamoorthy [20]	1.59	29	13.49	26	40.69	24
Kolisch [15]	1.29	27	13.53	27	41.84	25
Hartmann [8]	1.12	25	12.74	19	42.25	26
Kolisch [14]	1.00	24	14.30	28	43.05	27
Kolisch [14]	1.48	28	15.17	29	47.61	28
Klein [12]	0.17	10	12.03	14	-	
Baar, Brucker and Knust [3]	0.44	18	13.48	25	-	

representation, the thus updated activities all start within a particular time interval. The other RK components are not left unchanged, but are updated as follows. We subtract the constant value n from all RK values lower than p_{min}, and add the same constant to all values higher than p_{max}. Doing this preserves the priority structure since the activities outside the interval $[p_{min}, p_{max}]$ are unaffected by the forces. Table 1 reveals that this leads to an additional improvement for the J60 (from 11.43% to 11.29%), J90 (from 11.05% to 10.89%) and J120 (from 34.21% to 33.98%) instances.

3.2 Comparison with the State-of-the-Art Heuristics

To be able to compare procedures for the RCPSP, [10] presented a methodology in which all procedures can be tested on the PSPLIB datasets by using the number of generated schedules as a stop condition. Based on the methodology they also report state-of-the-art results. In [18] an update is given of these results. In table 5 we compare our algorithm with these results for the datasets J30, J60 and J120 respectively and for a stop condition of 5,000 schedules. In "Avg.Dev.Lb" we report the average deviation from the optimal solution for J30 or from the critical path based lowerbound for J60 and J120. The procedures are ranked according to their performance for the dataset J120. As this ranking slightly differs from the ranking for J30 and J60, we also provide a rank order in the column "Rank". The table reveals that the EM algorithm performs consistently well over all problem sets. Furthermore, the procedures that can outperform the EM procedure are hybrid heuristics. Consequently, we believe that the promising results might contribute to the further development of electromagnetism, possibly in combination (hybridization) with principles from other meta-heuristics.

4 Conclusions

This paper reports on results for the application of a new meta-heuristic procedure for solving combinatorial optimization problems. The procedure is a population-based method that is developed originally for optimizing unconstrained continuous functions based on an analogy with the electromagnetism theory. We illustrate the effective extension of this electromagnetism meta-heuristic to the well-known RCPSP.

The computational results show that the procedure produces consistently good results, compared to the state-of-the-art heuristics in the literature. Furthermore, all procedures that outperform the EM procedure are hybrid heuristics based on principles borrowed from various meta-heuristic approaches. Hence, we believe that the incorporation of ideas from EM in hybrid frameworks might contribute to the development of better meta-heuristic techniques. In the future we want to improve the performance of the EM heuristic for solving combinatorial problems by adding principles from other meta-heuristic techniques.

References

[1] Alcaraz, J., Maroto, C.: A robust genetic algorithm for resource allocation in project scheduling, Annals of Operations Research, 102, 83-109 (2001).

[2] Alcaraz, J., Maroto, C., Ruiz, R.: Improving the performance of genetic algorithms for the RCOS problems. Proceedings of the Ninth International Workshop on Project Management and Scheduling, 40-43, Nancy (2004)

[3] Baar, T., Brucker, P., Knust, S.: Tabu-search algorithms and lower bounds for the resource-constrained project scheduling problem, Meta-heuristics: Advances and trends in local search paradigms for optimization, 1-8 (1998).

[4] Birbil, S.I., Fang, S.C.: An electromagnetism-like mechanism for global optimization, Journal of Global Optimization 25 263-282 (2003).

[5] Bouleimen, K., Lecocq, H.: A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version, European Journal of Operational Research, 149, 268-281 (2003).

[6] Coelho, J., Tavares, L.: Comparative analysis of meta-heuricstics for the resource constrained project scheduling problem, Technical report, Department of Civil Engineering, Instituto Superior Tecnico, Portugal (2003).

[7] Debels, D., De Reyck, B., Leus, R., Vanhoucke, M.: A scatter-search meta-heuristic for the resource-constrained project scheduling problem, European Journal of Operational Research, forthcoming.

[8] Hartmann, S.: A competitive genetic algorithm for the resource-constrained project scheduling, Naval Research Logistics, 45, 733-750 (1998).

[9] Hartmann, S.: A self-adapting genetic algorithm for project scheduling under resource constraints, Naval Research Logistics, 49, 433-448 (2002).

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

[11] Klein, R.: Project scheduling with time-varying resource constraints. International Journal of Production Research, 38 (16): 3937-3952 (2000).

[12] Kochetov, Y., and Stolyar, A.: Evolutionary local search with variable neighbourhood for the resource constrained project scheduling problem, Proceedings of the 3rd International Workshop of Computer Science and Information Technologies (2003).

[13] Kolisch, R.: Project scheduling under resource constraints - Efficient heuristics for several problem classes, Physica (1995).

[14] Kolisch, R.: Serial and parallel resource-constrained project scheduling methods revisited: theory and computation, European Journal of Operational Research, 43, 23-40 (1996).

[15] Kolisch, R.: Efficient priority rules for the resource-constrained project scheduling problem, Journal of Operations Management, 14, 179-192 (1996).

[16] Kolisch, R., Drexl, A.: Adaptive search for solving hard project scheduling problems, Naval Research Logistics, 43, 23-40 (1996).

[17] Kolisch, R., Hartmann, S.; Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis. In: Weglarz, J. (Ed.), Project Scheduling - Recent Models, Algorithms and Applications, Kluwer Academic Publishers, Boston, pp. 147-178 (1999).

[18] Kolisch, R., Hartmann, S.: Experimental investigation of Heuristics for resource-constrained project scheduling: an update, working paper, Technical University of Munich (2004).

[19] Kolisch, R., Sprecher, A.: PSPLIB - A project scheduling library, European Journal of Operational Research, 96 205-216 (1997).
[20] Leon V. J., Ramamoorthy, B.: Strength and adaptability of problem-space based neighbourhoods for resource-constrained scheduling, OR Spektrum, 17, 173-182 (1995).
[21] Li, K.Y., Willis, R.J.: An iterative scheduling technique for resource-constrained project scheduling, European Journal of Operational Research, 56, 370-379 (1992).
[22] Merkle, D., Middendorf, M., Schmeck, H.: Ant colony optimization for resource constrained project scheduling, IEEE Transaction on Evolutionary Computation, 6(4), 333-346 (2002).
[23] Nonobe, K., Ibaraki, T.: Formulation and tabu search algorithm for the resource constrained project scheduling problem (RCPSP). In: Ribeiro, C.C., Hansen, P. (Eds.), Essays and Surveys in Meta-heuristics, Kluwer Academic Publishers, Boston, pp. 557-588 (2002).
[24] Schirmer, A.: Case-based reasoning and improved adaptive search for project scheduling, Naval Research Logistics, 47, 201-222 (2000).
[25] Sprecher, A.: Scheduling resource-constrained projects competitively at modest resource requirements, Management Science, 46 710-723 (2000).
[26] Tormos, P., Lova, A.: A competitive heuristic solution technique for resource-constrained project scheduling, Annals of Operations Research, 102, 65-81 (2001).
[27] Tormos, P., Lova, A.: An efficient multi-pass heuristic for project scheduling with constrained resources, International Journal of Production Research, 41, 1071-1086 (2003).
[28] Tormos, P., and Lova, A.: Integrating heuristics for resource constrained project scheduling: One step forward, Technical report, Department of Statistics and Operations Research, Universidad Politecnica de Valencia (2003).
[29] Valls, V., Quintanilla, S., Ballestin, F.: Resource-constrained project scheduling: a critical activity reordering heuristic, European Journal of Operational Research, 149, 282-301 (2003).
[30] Valls, V., Ballestin, F., Quintanilla, S.: A population-based approach to the resource-constrained project scheduling problem, Annals of Operations Research, 131, 305-324 (2004).
[31] Valls, V., Ballestin, F., Quintanilla, S.: A hybrid genetic algorithm for the Resource-constrained project scheduling problem with the peak crossover operator, Eighth International Workshop on Project Management and Scheduling, 368-371 (2002).
[32] Valls, V., Ballestin, F.: Quintanilla, S.: Justification and RCPSP: A technique that pays, European Journal of Operational Research, Forthcoming.