INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2013; 3(4):543-561

ROBUST RESOURCE-CONSTRAINED PROJECT SCHEDULING WITH UNCERTAIN-BUT-BOUNDED ACTIVITY DURATIONS AND CASH FLOWS
II. SOUNDS OF SILENCE: A NEW SAMPLING-BASED HYBRID PRIMARY-SECONDARY CRITERIA HARMONY SEARCH METAHEURISTIC

S. Danka*, †
Department of Structural Engineering, University of Pécs H-7624 Pécs, Boszorkány u. 2. Hungary

ABSTRACT

In this paper, we present a new idea for robust project scheduling combined with a cost-oriented uncertainty investigation. The result of the new approach is a makespan minimal robust proactive schedule, which is immune against the uncertainties in the activity durations and which can be evaluated from a cost-oriented point of view on the set of the uncertain-but-bounded duration and cost parameters using a sampling-based approximation. In this paper, we assume that the sources of uncertainty are the variability of the activity durations and the cash flow values, and present an appropriate hybrid method, which is a combination of mathematical programming, metaheuristic and sampling-based elements, to cope with this "uncertainty in uncertainty" like real problem.

Received: 10 April 2013; Accepted: 6 October 2013

KEY WORDS: project scheduling; scheduling under uncertainty; robustness

1. INTRODUCTION

In the real-word project scheduling problems, the "optimal" performance obtained using conventional deterministic methods can be dramatically degraded in the presence of sources

*Corresponding author: S. Danka, Department of Structural Engineering, University of Pécs H-7624 Pécs, Boszorkány u. 2. Hungary
†E-mail address: sandordanka@gmail.com. (S. Danka)

of uncertainty. Traditionally, project schedule uncertainty has been addressed by considering the uncertainty related to activity duration. In general, there are two approaches to dealing with uncertainty in a scheduling environment (Davenport and Beck [1]; Herroelen and Leus [2], and Vonder, et al. [3]): proactive and reactive scheduling. *Proactive scheduling* constructs a predictive schedule that accounts for statistical knowledge of uncertainty. The consideration of uncertainty information is used to make the predictive schedule more *robust*, i.e., insensitive to disruptions. *Reactive scheduling* involves revising or reoptimizing a schedule when an unexpected event occurs. At one extreme, reactive scheduling may not be based on a predictive schedule at all: allocation and scheduling decisions take place dynamically in order to account for disruptions as they occur. A less extreme approach is to reschedule when schedule breakage occurs, either by completely regenerating a new schedule or by repairing an existing predictive schedule to take into account the current state of the system.

In a recent paper, Danka [4] presented a new primary-secondary criteria approach for the resource-constrained project-scheduling problem (RCPSP) with uncertain-but-bounded activity durations and cash flows. In this paper, one of the most important conclusions was the following: without an appropriate hybrid method, which is combination mathematical programming, metaheuristic and sampling-based elements, we cannot cope with this "uncertainty in uncertainty" like problem. In the present paper, it will be shown that the originally time oriented "Sounds of Silence" harmony search metaheuristic developed by Csébfalvi et al. [5-7] for a wide range of different RCPS problems, with straightforward modifications can be used to solve the RCPS problem with uncertain activity durations and cash flows (RCPSP-UD-UC). The algorithm, as a new member of the Sounds of Silence (SoS) family produces optimal "robust" proactive schedules, which are immune against uncertainties in the activity durations. The presented robust schedule searching heuristic is based on a "forbidden set" oriented reformulation of the originally time oriented algorithm. In the presented algorithm, it is assumed that each activity duration and each cash flow value is an uncertain-but-bounded parameter, which can be characterized by its optimistic and pessimistic estimations. The primary optimality criterion is defined as a linear combination of the optimistic and pessimistic resource-feasible makespans, where the weights are able to describe the personal preferences of the project manager. The evaluation of a given robust schedule is based on the investigation of variability of the makespan as a primary and the net present value (*NPV*) as secondary criterion on the set of randomly generated scenarios given by a sampling-on-sampling-like process. In the simulation phase, the presented uncertain-but-bounded approach can be replaced by a possibilistic (membership function oriented) or probabilistic (density function oriented) approach, because the optimization model is insensitive to the "real meaning" of the optimistic and pessimistic estimations. In this paper, in the simulation phase a uniform random number generator was used to generate the uncertain parameters of the scenarios. Naturally, this simple approach can be replaced by more sophisticated parameter estimation process, but according to our experiences, the simulation process not so sensitive to the applied parameter generation method. In order to illustrate the efficiency and stability of the proposed Sounds of Silence (SoS) metaheuristic we present detailed computational results for a larger and challenging project instance borrowed from Golenko-Ginzburg and Gonic [8] and discussed by several authors in the literature. The presented reproducible results can be used for testing the quality of exact and heuristic solution procedures to be developed in the future in this area. The computational results reveal the fact that the Sounds of Silence (SoS) is fast, effective and robust algorithm.

2. PROBLEM FORMULATION

The theoretical description of the investigated problem, according to the applied primary-secondary criteria approach may be the following: The project consists of N activities $i \in \{1,2,...,N\}$ with nonpreemptable duration of D_i periods. In the traditional approach, it is assumed that each activity duration is a crisp value. Naturally, in the project-planning phase this assumption may be far from the reality. Imagine, for example, a new R&D project with several more or less new activities and an extremely long planning horizon.
Furthermore, activity $i = 0\,(\,i = N+1\,)$ is defined to be the unique dummy source (sink) with zero duration.

The activities are interrelated by precedence and resource constraints:
Precedence constraints force an activity not to be started before all its predecessors are finished. Let $\mathbf{NR} = \{ i \rightarrow j \mid i \neq j, i \in \{0,1,\ldots,N\}, j \in \{1,2,\ldots,N+1\}\}$ denote the set of immediate predecessor-successor relations (network relations).

Resource constraints arise as follows: In order to be processed, activity i requires R_{ir} units of resource type $r \in \{1,...,R\}$ during every period of its duration. Since resource r, $r \in \{1, ...,R\}$ is only available with the constant period availability of R_r units for each period, activities might not be scheduled at their earliest (network-feasible) start time but later. Let $\overline{T}$ denote the resource-constrained project's makespan and fix the position of the dummy sink in $\overline{T}+1$.

Without loss of generality, let α define the crisp discount rate in the planning horizon. Naturally, for a long-range project this assumption may be far from the reality, but methodological point of view, it can be replaced by a not necessarily continuous time function: $\alpha = \boldsymbol{\alpha(t)}$, $t \in \{0,\ldots,\overline{T}\}$ without difficulty. Let C_i, $i \in \{1,2,...,N\}$ denote the cash flow connected to activity i. By definition, the cash flow C_i, $i \in \{1,2,...,N\}$ may be negative, zero or positive and it is evaluated at the completion time of activity i. This assumption may be replaced by a more realistic one introducing dummy activities with zero duration and resource requirements as cash flow events, which connected to the real activities with predecessor-successor relations. Naturally, the essence of our model is not effected by this event oriented modification, which methodological point of view similar to hammock activity handling Csébfalvi and Csébfalvi [9].

The traditional RCPSP-NPV problem can be written as follows:

$$\boldsymbol{max}\left[NPV = \sum_{i=1}^{N} \sum_{t \in T_i} C_{it} * X_{it} \right] = NPV^*, \qquad (1)$$

$$X_i + D_i \le X_j,\ i \to j \in \mathbf{NR} \tag{2}$$

$$X_{N+1} = \overline{T} + 1, \tag{3}$$

$$X_i = \sum_{t \in T_i} X_{it} * t,\ T_i = \left\{\underline{X}_i, \underline{X}_i + 1, \dots, \overline{X}_i\right\},\ i \in \left\{1,2,\dots,N\right\}, \tag{4}$$

$$\sum_{t \in T_i} X_{it} = 1,\ X_{it} \in \left\{0,1\right\},\ i \in \left\{1,2,\dots,N\right\}, \tag{5}$$

$$A_t = \left\{i \mid X_i \le t < X_i + D_i, i \in \left\{1,2,\dots,N\right\}\right\},\ t \in \left\{1,2,\dots,T\right\} \tag{6}$$

$$U_{tr} = \sum_{i \in A_t} R_{ir},\ t \in \left\{1,2,\dots,T\right\},\ t \in \left\{1,2,\dots,T\right\},\ r \in \left\{1,2,\dots,R\right\} \tag{7}$$

$$U_{tr} \le R_r,\ t \in \left\{1,2,\dots,T\right\},\ r \in \left\{1,2,\dots,R\right\}, \tag{8}$$

$$C_{it} = C_i * e^{-\alpha(t + D_i - 1)},\ i \in \left\{1,2,\dots,N\right\},\ t \in T_i, \tag{9}$$

$$X_{it} \in \left\{0,1\right\},\ t \in T_i,\ i \in \left\{1,2,\dots,N\right\} \tag{10}$$

The binary decision variable set (10) specifies the possible starting times for each activity. By definition, the cash flow C_i connected to activity i, $i \in \left\{1,2,\dots,N\right\}$ may be negative, zero or positive and it is evaluated at its completion time. The discount rate is denoted by α. Objective (1) maximizes the discounted value of all cash flows that occur during the life of the project. Note that early schedules do not necessarily maximize the NPV of cash flows. Constraints (2) represent the precedence relations. In constraint (3) the resource-constrained project's makespan T can be replaced by its estimated upper bound. Constraints (4-5) ensure that each activity i, $i \in \left\{1,2,\dots,N\right\}$ has exactly one starting time within its time window $T_i = \left\{\underline{X}_i, \underline{X}_i + 1, \dots, \overline{X}_i\right\}$ where $\underline{X}_i$ ($\overline{X}_i$) is the early (late) starting time for activity i according to the precedence constraints and the latest project completion time $\overline{T}$. Constraints (6-8) ensure that resources allocated to activities at any time during the project do not exceed resource availabilities. Constraint set (9) for each activity describes the change of the cash flow in the function of the completion time.

The core element of the sampling-based cost-oriented schedule evaluation is a MILP problem, in which we try to maximize the NPV fixing the activity durations and the cash-flow values according to the generated random numbers. Generally, the solution of a MILP problem is a costly operation. When the MILP problem is a core element of a simulation process, the total time requirement of the MILP problem solutions may be a critical factor of the simulation, which may degrade the quality of the sample-based approximation.

Fortunatelly, according to the applied implicit resource-constraint handling, the constraint set of the MILP contains only precedence constraints, which consist of the original predecessor-successor relations and the inserted resource-conflict repairing relations.

Replacing the standard precedence constraints

$$S_i + D_i \le S_j\,,\ i \to j \in \mathbf{NR} \cup \mathbf{RR}^* \subset \mathbf{RR}\,, \tag{11}$$

with a totally unimodular (TU) formulation, the resource-constraint-free net present value problem ($NPVP$) can be solved in polynomial time as a LP problem (see Pritsker et al. [10]):

$$\boldsymbol{max}\left[NPV = \sum_{i=1}^{N} \sum_{t \in T_i} C_{it} * X_{it} \right] = NPV^*\,, \tag{12}$$

$$\sum_{p=T_i}^{\overline{X}_i} X_{ip} + \sum_{q=\underline{X}_p}^{T_i + D_i - 1} X_{jq} \le 1\,,\ T_i \in \left\{ \underline{X}_i, \underline{X}_i + 1, \ldots, \overline{X}_i \right\},\ i \to j \in PS \cup RR\,, \tag{13}$$

$$X_{N+1} = \overline{T} + 1\,, \tag{14}$$

$$\sum_{t \in T_i} X_{it} = 1\,,\ X_{it} \in \{0,1\},\ i \in \{1,2,\ldots,N\}, \tag{15}$$

$$\sum_{t \in T_i} X_{it} = 1\,,\ X_{it} \in \{0,1\},\ i \in \{1,2,\ldots,N\}, \tag{16}$$

$$X_{it} \in \{0,1\},\ t \in T_i\,,\ i \in \{1,2,\ldots,N\}, \tag{17}$$

Objective (12) maximizes the discounted value of all cash flows that occur during the life of the project. Note that early schedules do not necessarily maximize the NPV of cash flows. Constraints (13) represent the "strong" precedence relations. In constraint (14) the resource-constrained project's makespan T can be replaced by its estimated upper bound. Constraints (18) ensure that each activity i , $i \in \{1,2,\ldots,N\}$ has exactly one starting time within its time window $T_i = \left\{ \underline{X}_i, \underline{X}_i + 1, \ldots, \overline{X}_i \right\}$ where $\underline{X}_i$ ($\overline{X}_i$) is the early (late) starting time for activity i according to the precedence constraints and the latest project completion time $\overline{T}$. Constraint set (15) describes for each activity the change of the cash flow in the function of the completion time. The binary decision variable set (16) specifies the possible starting times for each activity. Using a fast interior-point-solver [11-12] the modified LP problem can be solved nearly 100 times faster then with a traditional simplex solver.

In the case of the RCPSP-UD-UC model, we have to assume, that

- Each activity duration D_i , $i \in \{1, 2,\ldots,N\}$ is a discrete (positive) uncertain-but-bounded parameter:

$$D_i \in \{ A_i, A_i + 1, \ldots, B_i \}, \tag{18}$$

where A_i and B_i are the optimistic and pessimistic estimations of D_i, respectively.

- Each activity cash flow C_i, $i \in \{ 1, 2, \ldots, N \}$ is a continuous (positive or negative) uncertain-but-bounded parameter:

$$C_i \in \left[\underline{C}_i, \overline{C}_i \right], \tag{19}$$

- After that, e have to generate a resource-conflict repairing relation set RR, which repairs all visible or hidden resource usage conflicts for each

$$D = \{ D_1, \ldots, D_N \},\ D_i \in \{ A_i, A_i + 1, \ldots, B_i \},\ i \in \{ 1, 2, \ldots, N \} \tag{20}$$

scenario such a way that on the makespan minimal scenario set for each

$$C = \{ C_1, \ldots, C_N \},\ C_i \in \left[\underline{C}_i, \overline{C}_i \right],\ i \in \{ 1, 2, \ldots, N \} \tag{21}$$

cash flow assignment of the schedule is "some how" optimal.

Naturally it is an open and very hard question, that how could we define the optimality in this case from managerial point of view. In this paper, we introduce a very simple, easy-to-understand measure to characterize a schedule (see Figure 1):

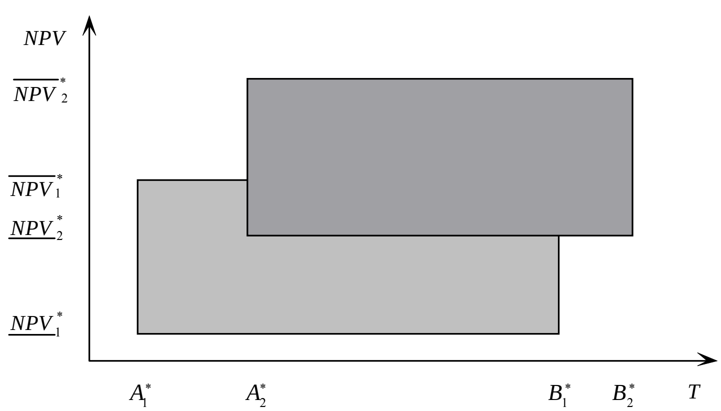

Figure 1. A simple measure for RCPSP-UD-UC

Figure 1 is a good visualization of a dilemma, that which schedule would be the better from managerial point of view. Naturally, the answer depends on the habit of the project manager.

3. ALGORITHM DESCRIPTION

Harmony search (HS) algorithm was recently developed by Lee and Geem [12] in an analogy with music improvisation process where music players improvise to obtain better harmony. In HS, the optimization problem is specified as follows:

$$max\left\{ f(X) \mid X = \left\{ X_i \mid \underline{X}_i \le X_i \le \overline{X}_i, i \in \{1,2,...,N\} \right\} \right\}. \tag{22}$$

In the language of music, $\boldsymbol{X}$ is a melody, which aesthetic value is represented by $f(\boldsymbol{X})$. Namely, the higher the value $f(\boldsymbol{X})$, the higher the quality of the melody is. In the band, the number of musicians is N, and musician i, $i = \{1,2,\dots,N\}$ is responsible for sound X_i. The improvisation process is driven by two parameters: (1) According to the repertoire consideration rate (RCR), each musician is choosing a sound from his/here repertoire with probability RCR, or a totally random value with probability $(1 - RCR)$; (2) According to the sound adjusting rate (SAR), the sound, selected from his/here repertoire, will be modified with probability SAR. The algorithm starts with a totally random "repertoire upload" phase, after that, the band begins to improvise. During the improvisations, when a new melody is better than the worst in the repertoire, the worst will be replaced by the better one. Naturally, the two most important parameters of HS algorithm are the repertoire size and the number of improvisations. The HS algorithm is an "explicit" one, because it operates directly on the sounds. In the case of RCPSP, we can only define an "implicit" algorithm, and without introducing a "conductor", we cannot manage the problem efficiently. In the world of music, the resource profiles form a "polyphonic melody". Therefore, assuming that in every phrase only the "high sounds" are audible, the transformed problem will be the following: find the shortest "Sounds of Silence" melody by improvisation! Naturally, the "high sound" in music is analogous to the overload in scheduling. In the language of music, the RCPSP can be summarized as follows: (1) the band consists of N musicians; (2) the polyphonic melody consists of R phrases and N polyphonic sounds; (3) each $i \in \{1,2,...,N\}$ musician is responsible for exactly one polyphonic sound; (4) each $i \in \{1,2,...,N\}$ polyphonic sound is characterized by the set of the following elements: $\left\{ X_i, D_i, \left\{ R_{ir} \mid r \in \{1,2,...,R\} \right\} \right\}$; the polyphonic sounds (musicians) form a partially ordered set according to the precedence (predecessor-successor) relations; (5) each $r \in \{1,2,...,R\}$ phrase is additive for the simultaneous sounds; (6) in each phrase only the high sounds are audible: $\left\{ U_{tr} \mid U_{tr} > R_r, t = 1,2,...,T \right\}$; (7) in each repertoire uploading (improvisation) step, each $i \in \{1,2,...,N\}$ musician has the right to present (modify) an idea $IP_i \in [-1,+1]$ about X_i where a large positive (negative) value means that the musician want to enter into the melody as early (late) as possible; (8) in the repertoire uploading phase the "musicians" improvise freely, $IP_i \leftarrow \boldsymbol{RandomGauss}(0,1)$, where function $\eta \leftarrow \boldsymbol{RandomGauss}(\mu,\sigma)$ generates random numbers from a truncated ($-1 \le \eta \le 1$) normal distribution with mean μ and standard deviation σ (9) in the improvisation phase the "freedom of imaginations" is decreasing step by step, $IP_i \leftarrow \boldsymbol{RandomGauss}(IP_i, \sigma)$, where standard deviation σ is a decreasing function of the progress (see figures 3 and 4); (10) each of the possible decisions of the harmony searching process (melody selection and idea-driven melody construction) is the conductor's responsibility; and (11) the band try to find the shortest "Sounds of Silence" melody by improvisation.

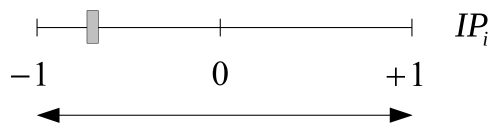

Figure 2. An idea IP_i about the "best" position

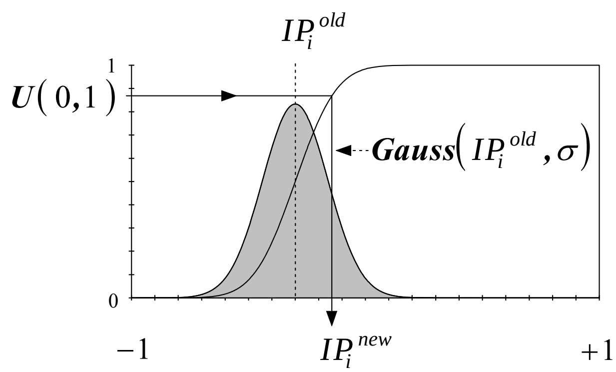

Figure 3. Perturbation of IP_i

The conductor solves a linear programming (LP) problem to balance the effect of the more or less opposite ideas about a shorter "Sounds of Silence" melody. The LP problem, which maximizes the satisfaction of the musicians with the sound positions, is the following:

$$\boldsymbol{min}\left[\sum_{i=1}^{N} IP_i * X_i\right], \tag{23}$$

$$X_i + D_i \leq X_j,\ i \rightarrow j \in PS, \tag{24}$$

$$\underline{X}_i \leq X_i \leq \overline{X}_i,\ i \in \{1,2,\ldots,N\}. \tag{25}$$

The result of the optimization is a schedule (melody) which is used by the conductor to define the final starting (entering) order of the sounds (musicians). The conductor generate a soundless melody by taking the selected sounds one by one in the given order and scheduling them at the earliest (latest) feasible start time. After that, using the well-known forward-backward improvement (FBI) methods (see, for example, Tormos and Lova [13]) the conductor tries to improve the quality of the generated melody. Naturally, the conductor memorizes the shortest feasible melody found so far.

The conflict repairing version of the Sounds of Silence algorithm is based on the forbidden set concept. In the conflict repairing version, the primary variables are conflict repairing relations, and a solution will be a makespan minimal resource-feasible solution set, in which every movable activity can be shifted without affecting the resource feasibility. In the traditional "time oriented" model the primary variables are starting times, therefore an activity shift may be able to destroy the resource feasibility.

The makespan minimal solutions of the conflict repairing model are immune against the activity movements, so we can introduce a not necessarily regular secondary performance measure to select the "best" makespan minimal resource feasible solution from the generated solution sets. In the Sounds of Silence algorithm, according to the applied replacement strategy (whenever the algorithm obtains a solution superior to the worst solution of the current population, the worst solution will be replaced by the better one) the quality of the population is increasing step by step. According to the progress of the searching process, the size of the makespan minimal subset of the population is increasing. The larger the makespan minimal subset size, the higher the chance to get a good solution for the secondary criterion. It is well-known, that the crucial point of the conflict repairing model is the forbidden set computation.

In the Sounds of Silence algorithm the conductor using a simple (but fast and effective) "thumb rule" to decrease the time requirement of the forbidden set computation. In the forward-backward list scheduling process, the conductor (without explicit forbidden set computation) inserts a precedence relation $i \rightarrow j$ between an already scheduled activity i and the currently scheduled activity j whenever they are connected without lag ($S_j + D_j = S_j$). The result will be schedule without "visible" conflicts.

After that, the conductor (in exactly one step) repairs all of the hidden (invisible) conflicts, inserting always the "best" conflict repairing relation for each forbidden set. In this context "best" means a relation $i \rightarrow j$ between two forbidden set members for which the lag ($S_j - S_i - D_i$) is maximal. In the language of music, the result of the conflict repairing process will be a robust (flexible)

The "Sounds of Silence" melody, in which the musicians have some freedom to enter to the performance without affecting the aesthetic value of the composition. Naturally, when we introduce a secondary criterion (in our case, for example, the *NPV* measure), for which the aesthetic value is a function of the starting times, the freedom of the musicians totally disappears.

3. COMPUTATIONAL EXPERIMENTS

The SoS for RCPSP-UD-UC problems has been programmed in Compaq Visual Fortran® Version 6.5. The algorithm, as a DLL, was built into the *ProMan* system (Visual Basic® Version 6.0) developed by Ghobadian and Csébfalvi [14]. To solve the LP problems a fast state-of-the-art primal-dual interior point solver, namely the DLL version of BPMPD developed by Mészáros was used. Naturally, this solver can be replaced by any other commercial (academic) LP solver. The computational results were obtained by running ProMan on a 1.8 GHz Pentium IV IBM PC with 256 MB of memory under Microsoft Windows XP® operation system.

We run the RCPSP-UD-UC specific SoS algorithm 30 times independently with four different settings and randomly generated starting seeds:

$$\{G10P10, G10P50, G10P100, G10P500\},$$

where, in the short description of the sets, G is the number of generations and P is the population size (see Table 1-4). The increasing population size significantly increases the quality of the solutions, and significantly decreases the spreading of the solutions, where the spreading was measured by the nonparametric range function [16-21]. The quality of the robust makespan characteristics was measured by the relative percent error using the best (Cplex) solution as an etalon:

$$\boldsymbol{E(}A+B\boldsymbol{)} = 100 * \frac{\left(A+B-A^*-B^*\right)}{A^*+B^*}\%,$$

$$\boldsymbol{E(}A\boldsymbol{)} = 100 * \frac{\left(A-A^*\right)}{A^*}\%,$$

$$\boldsymbol{E(}B\boldsymbol{)} = 100 * \frac{\left(B-B^*\right)}{B^*}\%,$$

We presented the results of the simulation phase for $G10P500$ ($G10P500+S1000$). The *NPV* characteristics are good indicators of the quality of the robust schedules from a cost-oriented point of view (see Table 5-7). We have to note, that according to the robust nature of the Central Limit Theorem, it is a very rare event that the quality of a robust schedule is extremely good or extremely bad. This fact is well detectable when we compare the theoretical range given by the robust SoS with the much smaller sample-based range given by simulation. The results well illustrate the fact, that in a multi-objective decision environment, the decision-making may be a very hard problem and a makespan minimal solution not necessarily should be preferable in an uncertain situation. The results reveal the fact, that the problem-specific SoS is a fast and robust algorithm without approach-specific tunable-parameters.

Table 1: The initial data of the Golenko-Ginzburg and Gonik project

Run	*A+B*	*A*	*B*	*E(A+B)* %	*E(A)* %	*E(B)* %	*Time* *sec*

Cplex	**840**	**340**	**500**	**0.00**	**0.00**	**0.00**	**36000**
1	844	342	502	0.48	0.59	0.40	0.406
2	851	339	512	1.31	-0.29	2.40	0.312
3	855	349	506	1.79	2.65	1.20	0.469
4	856	340	516	1.90	0.00	3.20	0.233
5	857	343	514	2.02	0.88	2.80	0.311
6	858	341	517	2.14	0.29	3.40	0.387
7	858	348	510	2.14	2.35	2.00	0.362
8	859	347	512	2.26	2.06	2.40	0.449
9	860	343	517	2.38	0.88	3.40	0.341
10	860	344	516	2.38	1.18	3.20	0.436
11	860	345	515	2.38	1.47	3.00	0.404
12	860	350	510	2.38	2.94	2.00	0.374
13	860	351	509	2.38	3.24	1.80	0.285
14	861	348	513	2.50	2.35	2.60	0.329
15	861	357	504	2.50	5.00	0.80	0.480
16	862	350	512	2.62	2.94	2.40	0.363
17	862	355	507	2.62	4.41	1.40	0.297
18	863	338	525	2.74	-0.59	5.00	0.267
19	863	351	512	2.74	3.24	2.40	0.329
20	863	353	510	2.74	3.82	2.00	0.405
21	864	343	521	2.86	0.88	4.20	0.343
22	864	344	520	2.86	1.18	4.00	0.486
23	864	347	517	2.86	2.06	3.40	0.270
24	864	353	511	2.86	3.82	2.20	0.281
25	865	353	512	2.98	3.82	2.40	0.501
26	866	345	521	3.10	1.47	4.20	0.422
27	867	345	522	3.21	1.47	4.40	0.298
28	867	345	522	3.21	1.47	4.40	0.283
29	867	353	514	3.21	3.82	2.80	0.268
30	868	352	516	3.33	3.53	3.20	0.376
Mean	861	347	514	2.50	2.10	2.77	0.359
Range	24	19	23	2.86	5.59	4.60	0.268

Table 2: The Cplex solution and the results of 30 independent SoS runs (G10P50)

Run	*A+B*	*A*	*B*	*E(A+B)* %	*E(A)* %	*E(B)* %	*Time* sec

Cplex	**840**	**340**	**500**	**0.00**	**0.00**	**0.00**	**36000**
1	839	341	498	-0.12	0.29	-0.40	1.830
2	846	343	503	0.71	0.88	0.60	1.968
3	848	344	504	0.95	1.18	0.80	2.054
4	850	346	504	1.19	1.76	0.80	1.688
5	850	346	504	1.19	1.76	0.80	1.604
6	851	351	500	1.31	3.24	0.00	1.791
7	852	338	514	1.43	-0.59	2.80	1.795
8	853	340	513	1.55	0.00	2.60	1.803
9	853	348	505	1.55	2.35	1.00	1.854
10	854	351	503	1.67	3.24	0.60	2.092
11	854	351	503	1.67	3.24	0.60	1.911
12	855	341	514	1.79	0.29	2.80	2.045
13	855	346	509	1.79	1.76	1.80	1.624
14	856	339	517	1.90	-0.29	3.40	1.774
15	856	351	505	1.90	3.24	1.00	1.720
16	857	343	514	2.02	0.88	2.80	1.688
17	857	346	511	2.02	1.76	2.20	1.738
18	857	354	503	2.02	4.12	0.60	1.904
19	858	350	508	2.14	2.94	1.60	1.773
20	858	350	508	2.14	2.94	1.60	1.932
21	858	351	507	2.14	3.24	1.40	1.795
22	858	353	505	2.14	3.82	1.00	1.778
23	859	339	520	2.26	-0.29	4.00	1.682
24	859	348	511	2.26	2.35	2.20	2.064
25	860	344	516	2.38	1.18	3.20	1.983
26	860	350	510	2.38	2.94	2.00	2.014
27	860	351	509	2.38	3.24	1.80	1.878
28	861	352	509	2.50	3.53	1.80	1.924
29	863	355	508	2.74	4.41	1.60	1.803
30	863	355	508	2.74	4.41	1.60	1.918
Mean	857	348	509	1.83	2.13	1.62	1.848
Range	24	17	22	2.86	5.00	4.40	0.488

Table 3: The Cplex solution and the results of 30 independent SoS runs (G10P100)

Run	*A+B*	*A*	*B*	*E(A+B)*	*E(A)*	*E(B)*	*Time*

				%	%	%	*sec*
Cplex	**840**	**340**	**500**	**0.00**	**0.00**	**0.00**	**36000**
1	844	344	500	0.48	1.18	0.00	4.367
2	844	345	499	0.48	1.47	-0.20	4.029
3	851	337	514	1.31	-0.88	2.80	3.922
4	851	340	511	1.31	0.00	2.20	4.389
5	851	346	505	1.31	1.76	1.00	4.340
6	851	347	504	1.31	2.06	0.80	4.459
7	851	350	501	1.31	2.94	0.20	4.293
8	852	342	510	1.43	0.59	2.00	4.269
9	852	344	508	1.43	1.18	1.60	4.815
10	852	344	508	1.43	1.18	1.60	4.274
11	852	344	508	1.43	1.18	1.60	4.496
12	852	345	507	1.43	1.47	1.40	4.655
13	852	346	506	1.43	1.76	1.20	4.533
14	852	346	506	1.43	1.76	1.20	4.435
15	852	346	506	1.43	1.76	1.20	4.447
16	852	349	503	1.43	2.65	0.60	4.256
17	853	345	508	1.55	1.47	1.60	4.793
18	854	341	513	1.67	0.29	2.60	4.501
19	854	342	512	1.67	0.59	2.40	5.341
20	854	346	508	1.67	1.76	1.60	4.558
21	854	346	508	1.67	1.76	1.60	4.401
22	855	339	516	1.79	-0.29	3.20	4.421
23	855	346	509	1.79	1.76	1.80	4.676
24	855	348	507	1.79	2.35	1.40	6.729
25	855	348	507	1.79	2.35	1.40	4.795
26	855	352	503	1.79	3.53	0.60	4.341
27	856	347	509	1.90	2.06	1.80	4.125
28	856	347	509	1.90	2.06	1.80	4.466
29	857	352	505	2.02	3.53	1.00	4.149
30	858	342	516	2.14	0.59	3.20	4.484
Mean	853	345	508	1.52	1.53	1.51	4.525
Range	14	15	17	1.67	4.41	3.40	2.807

Table 4: The Cplex solution and the results of 30 independent SoS runs (G10P500)

Run	*A+B*	*A*	*B*	*E(A+B)* %	*E(A)* %	*E(B)* %	*Time* *sec*
Cplex	**840**	**340**	**500**	**0.00**	**0.00**	**0.00**	**36000**
1	842	343	499	0.24	0.88	-0.20	14.837
2	842	344	498	0.24	1.18	-0.40	14.052
3	849	336	513	1.07	-1.18	2.60	10.619
4	849	339	510	1.07	-0.29	2.00	11.370
5	849	345	504	1.07	1.47	0.80	11.074
6	849	346	503	1.07	1.76	0.60	10.183
7	849	349	500	1.07	2.65	0.00	12.011
8	850	341	509	1.19	0.29	1.80	14.221
9	850	343	507	1.19	0.88	1.40	15.079
10	850	343	507	1.19	0.88	1.40	12.035
11	850	343	507	1.19	0.88	1.40	12.778
12	850	344	506	1.19	1.18	1.20	14.927
13	850	345	505	1.19	1.47	1.00	14.964
14	850	345	505	1.19	1.47	1.00	14.067
15	850	345	505	1.19	1.47	1.00	12.428
16	850	348	502	1.19	2.35	0.40	11.311
17	851	344	507	1.31	1.18	1.40	15.839
18	852	340	512	1.43	0.00	2.40	10.258
19	852	341	511	1.43	0.29	2.20	11.416
20	852	345	507	1.43	1.47	1.40	12.505
21	852	345	507	1.43	1.47	1.40	13.296
22	853	338	515	1.55	-0.59	3.00	13.455
23	853	345	508	1.55	1.47	1.60	13.138
24	853	347	506	1.55	2.06	1.20	10.358
25	853	347	506	1.55	2.06	1.20	10.288
26	853	351	502	1.55	3.24	0.40	15.189
27	854	346	508	1.67	1.76	1.60	12.607
28	854	346	508	1.67	1.76	1.60	10.841
29	855	351	504	1.79	3.24	0.80	11.694
30	856	341	515	1.90	0.29	3.00	13.606
Mean	851	344	507	1.28	1.24	1.31	12.681
Range	14	15	17	1.67	4.41	3.40	5.656

Table5: The best Cplex solution and the ordered results for 30 independent SoS runs (G10P500)

i	Makespan Measures			i	Makespan Measures		
	A+B	*A*	*B*		*A+B*	*A*	*B*
Cplex	**840**	**340**	**500**		**840**	**340**	**500**
1	842	343	499	16	850	348	502
2	842	344	498	17	851	344	507
3	849	336	513	18	852	340	512
4	849	339	510	19	852	341	511
5	849	345	504	20	852	345	507
6	849	346	503	21	852	345	507
7	849	349	500	22	853	338	515
8	850	341	509	23	853	345	508
9	850	343	507	24	853	347	506
10	850	343	507	25	853	347	506
11	850	343	507	26	853	351	502
12	850	344	506	27	854	346	508
13	850	345	505	28	854	346	508
14	850	345	505	29	855	351	504
15	850	345	505	30	856	341	515

Table 6: The best Cplex solution and ordered result of the approximated solutions for 30 independent SoS runs (G10P500 + S1000)

i	Makespan Range		Net Present Value	
	$\underline{A}$	$\overline{B}$	$\underline{NPV}$	$\overline{NPV}$
SoS	**395**	**465**	**1579**	**2111**
1	393	475	1624	2222
2	395	471	1425	1932
3	395	477	1547	2145
4	397	472	1374	1828
5	398	468	1423	1923
6	398	476	1586	2166
7	398	478	1501	2002
8	398	486	1796	2438
9	399	471	1576	2095
10	399	476	1370	1884
11	399	476	1411	1895
12	400	475	1612	2147
13	401	474	1494	1983
14	402	469	1357	1760
15	402	472	1730	2267

Table 7: The best Cplex solution and ordered result of the approximated solutions for 30 independent SoS runs (G10P500 + S1000)

i	Makespan Range		Net Present Value	
	$\underline{A}$	$\overline{B}$	$\underline{NPV}$	$\overline{NPV}$
SoS	**395**	**465**	**1579**	**2111**
16	402	481	1473	1958
17	403	472	1489	1972
18	403	475	1419	1865
19	403	482	1451	1973
20	404	472	1488	1964
21	404	472	1585	2088
22	404	473	1571	2072
23	404	475	1200	1591
24	404	481	1497	1963
25	405	469	1574	2079
26	405	489	1385	1829
27	406	475	1319	1777
28	408	476	1384	1861
29	408	485	1448	1958
30	409	470	1489	1979

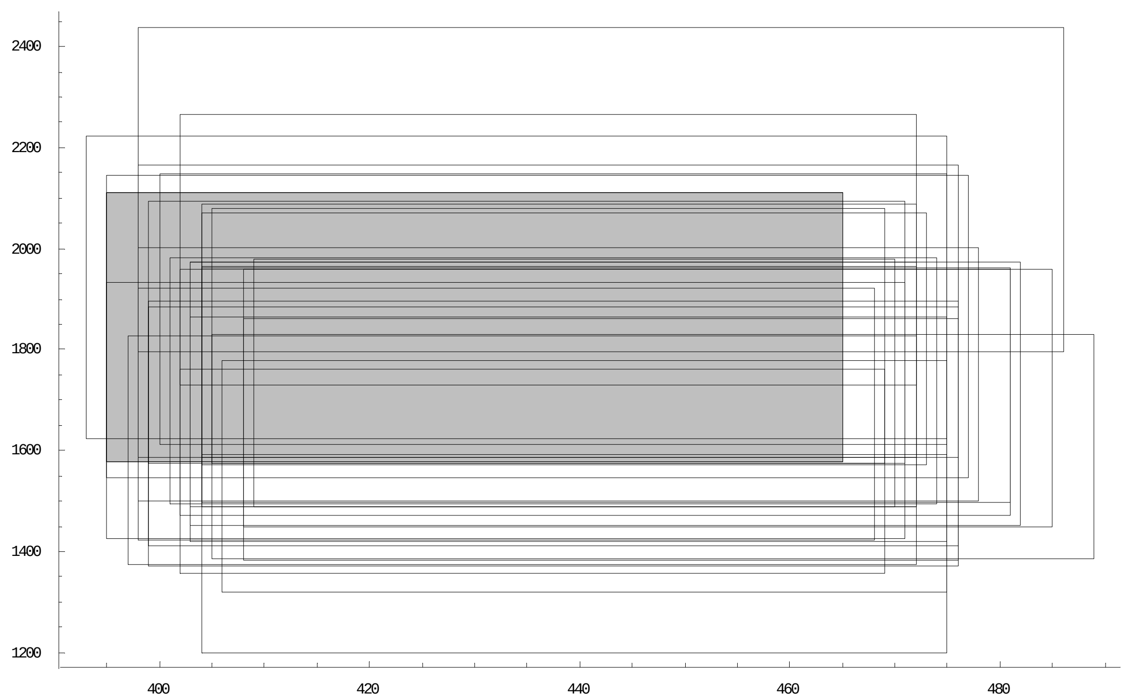

Figure 4. Visualization of the Cplex solution and the solutions of 30 independent SoS runs

5. CONCLUSONS

In this paper, we presented a new hybrid harmony search metaheuristic combined with sampling-based solution approximation for the resource-constrained project-scheduling problem (RCPSP) with uncertain-but-bounded activity durations and cash flows (RCPSP-UD-UC). The presented Sound of Silence (SoS) algorithm, which is an appropriate combination of mathematical programming, metaheuristic and sampling-based elements, is a straightforward modification of the originally time oriented “Sounds of Silence” harmony search metaheuristic developed for a wide range of different RCPS problems by Csébfalvi [20],by Csébfalvi and Láng [21],by Csébfalvi and Szendrői [22]. The algorithm, as a new member of the Sounds of Silence (SoS) family produces optimal “robust” proactive schedules, which are immune against the uncertainties in the activity durations.

The presented robust schedule searching heuristic is based on a “forbidden set” oriented reformulation of the originally time oriented algorithm. In the presented algorithm, it is assumed that each activity duration and each cash flow value is an uncertain-but-bounded parameter, which can be characterized by its optimistic and pessimistic estimations. The primary optimality criterion is defined as a linear combination of the optimistic and pessimistic resource-feasible makespans, where the weights are able to describe the personal preferences of the project manager.

The evaluation of a given robust schedule is based on the investigation of variability of the makespan as a primary and the net present value (NPV) as secondary criterion on the set of randomly generated scenarios given by a sampling-on-sampling-like process. In the simulation phase, the presented "uncertain-but-bounded" approach can be replaced by a possibilistic (membership function oriented) or probabilistic (density function oriented) approach, because the optimization model is insensitive to the “real meaning” of the optimistic and pessimistic estimations. In this paper, in the simulation phase a uniform random number generator was used to generate the uncertain parameters of the scenarios.

Naturally, this simple approach can be replaced by more sophisticated parameter estimation process, but according to our experiences, the simulation process not so sensitive to the applied parameter generation method. In order to illustrate the efficiency and stability of the proposed SoS metaheuristic we presented detailed computational results for a larger and challenging project instance borrowed from Golenko-Ginzburg and Gonic [8] and discussed by several authors in the literature.

The presented reproducible results can be used for testing the quality of exact and heuristic solution procedures to be developed in the future in this area. The computational results reveal the fact that the modified and extended SoS is fast, effective and robust algorithm, which is able to cope successfully with the project-scheduling problems when we replace the traditional crisp parameters with uncertain-but-bounded parameters.

REFERENCES

1. Davenport AJ, Beck JC. *A Survey of Techniques for Scheduling with Uncertainty*, Toronto, Canada, unpublished manuscript, 2002.

2. Herroelen W, Leus R. Project scheduling under uncertainty: Survey and research potentials, *Eur J Oper Res*, 2005; **165**: 289–306.
3. Van de Vonder S, Demeulemeester E, Herroelen W, Leus R. The use of buffers in project management: the trade-off between stability and makespan, *Int J Prod Econ,* 2005; **97**, 227–40.
4. Danka S, Robust resource-constrained project scheduling with uncertain-but-bounded activity durations and cash flows, I. A new sampling-based hybrid primary-secondary criteria approach, *Int J Optim Civil Eng*, 2013, **3** (4), 527-42.
5. Csébfalvi A, Csébfalvi G, Szendrői E. A harmony search metaheuristic for the resource-constrained project scheduling problem and its multi-mode version, In: Serifoglu FS, Bilge Ü (Eds.), *Project Management and Scheduling* 2008, Istanbul, Turkey, 2008, ISBN: 978-9944-62-618-7, 56–59.
6. Csébfalvi G, Eliezer, O, Láng B, Levi R. A conflict repairing harmony search metaheuristic and its application for bi-objective resource-constrained project scheduling problems, In: Serifoglu F.S., Bilge, Ü. (Eds.), *Project Management and Scheduling 2008*, Istanbul, Turkey, ISBN: 978-9944-62-618-7, 60–63.
7. Csébfalvi A, Szendrői E, An experimental investigation of the sounds of scilence metaheuristic for the multi-mode resource-constrained project scheduling with pre-optimized repertoire on the hardest MMLIB+SET, *Int J Optim Civil Eng,* 2012, **2** (4) 545–56.
8. Golenko-Ginzburg D, Gonik A, Stochastic network project scheduling with non-consumable limited resources, *Int J Prod Econ*, 1997, **48**, 29–37.
9. Csébfalvi A, Csébfalvi G, Hammock. Activities in Project Scheduling, In: Shrikant Panwalkar, Jian Li (Eds.), *Proceedings of the Sixteenth Annual Conference of POMS: "OM Frontiers: Winds of Change*, Chicago, USA, 2005, pp. 1–27.
10. Pritsker AAB, Watters LJ, Wolfe PM, Multi-project scheduling with limited resources: a zero-one programming approach. *Manag Sci* 1969, **16** (1) *Theory Series*, 93–108.
11. Csébfalvi A, Csébfalvi G, The Multidimensional 0-1 Knapsack Problem: A New Heuristic Algorithm Combined with 0-1 Linear Programming, In: Agostinho Rosa, Janusz Kacprzyk, Joaquim Filipe, António Dourado Correia (Eds.) ECTA 2011: *International Conference on Evolutionary Computation Theory and Applications and International Conference on Fuzzy Computation Theory and Applications*, SciTePress, 2011, (ISBN:978-989-8425-83-6), 203–207.
12. Lee KS, Geem ZW, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Comput Meth Appl Mech Eng,* 2005, **194**: 3902–3933.
13. Tormos P, Lova A, A competitive heuristic solution technique for resource-constrained project scheduling, *Ann Oper Res*, 2001, **102** 65–81.
14. Ghobadian A, and Csébfalvi G, Workshop on Developing Interactive Learning Material for Project Management, *Annual Meeting of Decision Sciences Institute, Boston*, USA, 1995.
15. Csébfalvi A, A theoretically correct resource usage visualization for the resource-constrained project scheduling problem, *Int J Optim Civil Eng*, 2012, **2**(2) 173–81.
16. Csébfalvi A, Kolmogorov-Smirnov test to tackle fair comparison of heuristic approaches in structural optimization, *Int J Optim Civil Eng*, 2012, **2** (1) 135–50.

17. Csébfalvi A, Csébfalvi G, Fair Comparison of Population-based Heuristic Approaches: The Evils of Competitive Testing, In: Agostinho Rosa, António Dourado, Kurosh Madani, Joaquim Filipe, Janusz Kacprzyk (Eds.), *IJCCI 2012 Proceedings of the 4th International Joint Conference on Computational Intelligence*, SciTePress, 2012. (ISBN:978-989-8565-33-4), 206–309.
18. Csébfalvi A, Which is the Best? A Statistically Correct Comparison of Heuristic Results in Structural Optimization, in B.H.V. Topping, (Editor), *Proceedings of the Eighth International Conference on Engineering Computational Technology*, Civil-Comp Press, Stirlingshire, UK, 2012. doi:10.4203/ccp.100.67, Paper 67.
19. Csébfalvi A, Pálné Schreiner J, A Net Present Value Oriented Hybrid Method to Optimize the Revenue of Geothermal Systems with Respect to Operation and Expansion, in Y. Tsompanakis, B.H.V. Topping, (Editors), *Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*", Civil-Comp Press, Stirlingshire, UK, 2011. doi:10.4203/ccp.97.21, Paper 21.
20. Csébfalvi A, A Unified Model for Resource-Constrained Project Scheduling Problem with Uncertain Activity Durations, *Int J Optim Civil Eng*, 2012, **2** (3), 1–15.
21. Csébfalvi A, Láng B. An improved hybrid method for the resource-constrained project scheduling problem with discounted cash flows: doi: 10.1556/Pollack.7.2012.1.13, *Pollack Periodica : an International Journal for Engineering and Information Sciences*, 2012, **7** (1), 135–46.
22. Csébfalvi A, Szendrői E. An Improved Hybrid Method for the Multi-Mode Resource-Constrained Project Scheduling Problem, In: B H V Topping (Eds.), *Proceedings of the Eighth International Conference on Engineering Computational Technology*, Civil-Comp Press, Stirling, UK, 2012, (ISBN:978-1-905088-55-3), Paper 50, pp. 1–13.