

An Orthogonal hybrid algorithm for the resource-constrained project scheduling problem

Zhiyu Huang

*Laboratory of complex systems and intelligence science
Institute of automation, Chinese academy of sciences
Beijing, 100080, China
huangzhiyu2004@163.com*

Abstract - As an effective method, the generation-based heuristic algorithm gets information from the history search at one generation, and then uses this information to generate some other solutions as the next generation so that a near optimum solution can be found. To do this effectively, how to represent a solution is a fundamental problem. After providing a new representation for a solution, a hybrid algorithm was given. The algorithm used the ideal of orthogonal design to make the solutions generating more reasonable, and used the ideal of Estimation of Distribution Algorithm to elicit the information about activity linking in a population so that the solutions with history linking information can be generated, and used the ideal of Scatter Search to keep the search process strolling in divers parts, and used the ideal of reverse schedule to improve the fitness of found solutions. The simulation results reveal the efficiency of the algorithm.

Index Terms - Scatter Search, Estimation of Distribution Algorithm, Reverse Schedule, Orthogonal Design, Solution Representation

I. INTRODUCTION

The resource-constrained project scheduling problem (RCPSP) can be given as follows:

Given are a set $N=\{0,1,\dots,n,n+1\}$ of $n+2$ activities and r renewable resources $1,\dots,r$. Fictitious activities 0 and $n+1$ correspond to the "project start" and to the "project end", respectively. A constant amount of R_k units of resources $k=1,\dots,r$ is available at any time. Activity i must processed for p_i time units, where preemption is not allowed. During this time period a constant amount of r_{ik} units of resource k is occupied. Furthermore, finish-start precedence relations are defined between the activities. The values R_k , p_i and r_{ik} are supposed to be non-negative integers. The objective is to determine starting times S_i for the activities $i=0, 1\dots n+1$ in such a way that the resources and precedence constraints are fulfilled and the makespan is minimized.

As a generalization of the classical job shop scheduling problem, RCPSP belongs to the class of NP-hard optimization problems. Therefore, heuristic solution procedures are reasonable when solving large problem instances. Heuristic algorithms may simply be part into two types: rule-based heuristic and metaheuristic (for detail cf. [1][2]). The rule-based heuristic method can be described as follows: Using Serial Schedule Generation Scheme (SSGS) or Parallel Schedule Generation Scheme (PSGS), at every decide stage,

an eligible activity based on the rule given in advance is selected to enter the partial schedule till all the activities are scheduled [4]. The main idea of metaheuristic methods is using the information elicited by searching process to guide the searching direction till a good enough solution is found or a stopping criterion is fulfilled. Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA) are normal metaheuristic methods, Scatter Search (SS) [5],[6] and Estimation of Distribution Algorithm (EDA) [7],[8] are used for RCPSP recently.

The searching procedure of **generation-based heuristic algorithm** is an iterated scan in the search space. At each scan, a population of solutions is found. In order to get a next population in which the solutions are better than the ones in the current population, what information is elicited to guide the next scan and how using the information to generate next population must be carefully considered. That is to say, taking which method to schedule is not important, using which way to get and translate information is the main considered issue.

II. EXTRA RELATION REPRESENTATION AND EXTRA RELATION SPACE

General speaking, there are three methods to represent a solution of RCPSP, that is Activity List (AL), Random-Key (RK) [4] and Schedule Scheme Representation (SSR) [9]. Let $X=\{x_1,\dots,x_n\}$ be the vector for the representation of real activities except the two fictitious activities 0 and $n+1$. As for the AL representation, x_i is the i -th activity to be scheduled, in other words, in AL representation, X is a permutation of the set of activities $N=\{1,\dots,n\}$. While for the RK representation, x_i is a privilege value for selecting activity i to be scheduled (Priority Rule Representation can also be regard as a special way of RK representation). Schedule Scheme Representation (SSR) [9] is very different from the others, which uses four relation sets (C, D, N, F) to represent a solution. Where C, D, N, F includes all the conjunctions, disjunctions relations, parallel relations and flexibility relation respectively.

Given a representation of AL or RK, uses one of schedule generation schemes (PSGS or SSGS) to generate a solution. As for SSR, a transitive orientation from (C, D, N, F) to (C', D', N', F') must be proceeded, and then uses the so called earliest start schedule to generate a solution.

When taking the ideal of Scatter Search into consideration, using AL and RK representations will have following difficulties:

1) the trouble of defining distance between solutions

The distance between a solution and itself should be zero, as two different representations of AL (RK) can be decoded into a same solution, it will be difficult to define a reasonable distance in the search space decided by AL (RK) representation. To overcome this trouble, an operation for AL (RK) must be taken so that one solution corresponds to only one AL (RK) [6]; even using this operation, there is another trouble, that is:

2) the trouble of selecting which schedule generation schemes and using of mixture schemes

For a given AL (RK) representation, using PSGS and SSGS may generate two different solutions. Unfortunately there is no theory to support which generation scheme is preferential. Recently there is a tide using mixture scheme to create solutions, in these methods the definition of distance is very difficult.

Schedule Scheme Representation is a competent representation as it can not use any schedule generation scheme to create a solution. Thanks to the theorem 1 of Reference [9], a SSR $(C', \emptyset, N', \emptyset)$ corresponds to only one solution and each solution corresponds to only one SSR. But as a special representation for branch-and-bound algorithm, it can not be used easily to define the distance between two solutions.

In order to overcome the difficulties mentioned above, inspired by Schedule Scheme Representation and the procedure of adding extra relations in the Reference [10], a new representation based on extra relation is given as follows.

A. Extra Relation Representation

Given a solution, using the following methods will build an Extra Relation Representation (ERR):

Adding Method: For arbitrary activities $i \neq j$, between which there is no conjunctions deduced by finish-start precedence relations and time window constraints, if the condition $F_i \leq S_j$ holds, a *normal extra relation* between activities i and j is added to build a representation

It can be proof that if adding all the extra relations normal to the project as the new finish-start constraints, scheduling the new project by earliest start schedule can get the same solution used for building extra relation representation [11].

Contrasting to the Schedule Scheme Representation (SSR), extra relation representation uses only a subset of conjunctions set C to represent a solution and the decoding procedures to a solution are same. As not using any schedule generation scheme to create a solution, ERR will overcome the difficulties of the AL and RK representation for SS. As considering only a subset of conjunctions, it provides a practical way to define the distance between two solutions. It should be noted that same as the SSR decoding a given ERR may get an unfeasible solution while decoding AL or RK representation will get feasible solution only.

B. Extra Relation Space and Distance defining

The finish-start precedence constraints may be described by a link matrix. Let $L = (l_{ij})_{i,j \in N}$ be the link matrix, where N is the set of activities, the element l_{ij} is defined by:

$$l_{ij} = \begin{cases} 1 & \text{if } i \rightarrow j \\ 0 & \text{if others} \end{cases} \quad (1)$$

The conjunctions in formula (1) are deduced by finish-start precedence relations and time window constraints. As the conjunction relation is a transitive relation, the finish-start precedence relations can be deduced as follows:

$$i \rightarrow j \wedge j \rightarrow k \Rightarrow i \rightarrow k \quad (2)$$

Let LF_i , ES_j be the last finish time of activity i and early start time of activity j , respectively. Using the following procedure can get a new conjunction relation.

$$LF_i \leq ES_j \Rightarrow i \rightarrow j \quad (3)$$

To get a link matrix, we can apply Floyd-Warshall algorithm to the matrix which is initialized by setting $l_{ij}=1$ if either there is a defined finish-start precedence relations, or the formula (3) is fulfilled and setting $l_{ij}=0$ for the other conditions. As a normal extra relation can be added only between two activities which have no any conjunctions, the position (i,j) may be used as an extra relation representation is decided by:

$$l_{ij} = 0 \wedge l_{ji} = 0 \quad (4)$$

Getting all the positions fulfilled the formula (4) and ordering these positions by a way given in advance, every extra relation representation can be wrote by a vector. In this way, all the ordered extra relation positions construct a search space, called by Extra Relation Space. Every ERR corresponds to a point in this space. Let $P = \{p_{ijk} | l_{ij} = 0 \wedge l_{ji} = 0, 1 \leq k \leq m\}$ be the ordered vector including all the positions which can be added extra relation, the subordinate index k in the element p_{ijk} corresponds to the position of extra relation $i \rightarrow j$ in P , an extra relation representation can be defined by $X = \{x_k\}$, $x_k = 0, 1$. The formula $x_k = 1$ means that an extra relation $i \rightarrow j$ defined by p_{ijk} in the k -th position of P is selected as one elements of an Extra Relation Representation. It is clear that there are two symmetrical extra relations in P , that is to say, if $i \rightarrow j$ is an extra relation, then $j \rightarrow i$ is an extra relation too.

Hereinafter, call X as the coordinate of an ERR in the Extra Relation Space defined by P , call p_{ijk} and the corresponding extra relation $i \rightarrow j$ as the k -th dimension of the search space indiscriminately.

It is clear that there are two symmetrical extra relations in P , that is to say, if $i \rightarrow j$ is an extra relation, then $j \rightarrow i$ is an extra relation too.

The distance can be defined by any way used for binary string similarity or dissimilarity metrics. In this paper the distance between two ERRs $X_1 = \{x_{1k}\}$ and $X_2 = \{x_{2k}\}$ is defined by:

$$|X_1 - X_2| = \frac{\sum_{k=1}^m |x_{1k} - x_{2k}|}{m}, x_{ik} = \{0, 1\}, 1 \leq k \leq m \quad (5)$$

The distance defined by formula (5) measures the difference between extra relations representing for the two solutions. The bigger distance means the more different extra relations are taken to represent the two solutions and the probability of the two solutions with different makespan is higher.

III. ORTHOGONAL HYBRID ALGORITHM

The orthogonal design is commonly used to design experiments in the chemical field. It provides a series of orthogonal arrays for a given number of factors and a given number of levels. It has been proved that the orthogonal design is optimal for additive model and quadratic model, and the selected combinations are good representatives for all the possible combinations [12]. Inspired by the Reference [13], an orthogonal algorithm was designed. The whole procedure of the hybrid algorithm is as follows:

TABLE 1. ORTHOGONAL HYBRID ALGORITHM

1.	Initialize (I) two populations for forward and backward directions of the project respectively
2.	Repeat
3.	For each direction population
4.	Cluster (A) the solutions in the population
5.	For each cluster
6.	Learn and Sample(E) the information given by the solutions in the cluster, and generate some ERRs(D), then Decode(B) these ERRs to solutions
7.	Select the solution with the best fitness in the new generating solutions and original solutions in the cluster, as the reference solution for the cluster, and then add the reference solution to the next population and remove other solutions from this cluster
8.	End For
9.	Local move (F) every new gain solution in the population
10.	End For
11.	Link (G) the solutions in forward (backward) population to the solutions in the backward (forward) population
12.	For each direction population
13.	Learn and Sample(E) the information given by the solutions in the population, and generate some ERRs, then Decode(B) these ERRs to solutions
14.	Local move(F) every new gain solution
15.	Add all the new solutions to the current population
16.	End For
17.	For each direction population
18.	Reverse(H) schedule every solution in the population, and add the new solution to the reverse population
19.	End For
20.	Until stop criteria (J) is fulfilled
21.	End

The detail procedure for every operation will be given in the following subsections.

A. Clustering

Using the distance defined by (5), we can part the population of solutions into some parts by k -mean or hierarchical cluster method. In this paper, I use k -mean cluster method to part the population into a given number clusters. As the procedure of k -mean cluster is very simple, at this place I omit the detail procedure.

B. Decoding for ERR

As has mentioned above, using *earliest start schedule* to decode an ERR may get an unfeasible solution. To overcome this trouble, first, I get a new partial order by applying some operations to a given partial order, and then use SSGS for AL to get a feasible solution. The procedure of decoding an ERR to a solution is as follows:

Let $PO = \{pr_1, \dots, pr_n\}$ be a permutation of $N = \{1, \dots, n\}$, $L = (l_{ij})_{i,j \in N}$ be the current link matrix; $P = \{p_{ijk} | l_{ij} = 0 \wedge l_{ji} = 0, 1 \leq k \leq m\}$ be Extra Relation Space and $X = \{x_k | x_k = 0, 1; 1 \leq k \leq m\}$ be an EER in the space.

TABLE 2. ALGORITHM.DECODING

Algorithm.Decomding(X, PO, P, L)
1. Set $k := 1$
2. Repeat
3. If $x_k = 1$ then
4. Set $l'_{ij} := p_{ijk}$
5. Search in PO , getting the value f and s so that $pr_f = i, pr_s = j$
6. If $f \leq s$ then
7. MoveActivityToLeft (f, PO, L)
8. MoveActivityToRight (s, PO, L)
9. Else
10. ExchangePosInOrder (s, f, PO, L)
11. End If
12. End If
13. Set $j := j + 1$
14. Until $j = m$
15. Using the method SSGS for AL to get a solution with partial order PO
16. Return the result feasible solution
17. END Decoding
Algorithm.MoveActivityToLeft(f, PO, L)
1. Move the activity in the position f to the left as far as possible until a precedence constraint is violated.
Algorithm.MoveActivityToRight(s, PO, L)
1. Move the activity in the position s to the right as far as possible until a precedence constraint is violated
Algorithm.ExchangePosInOrder(f, s, PO, L)
1. For the activities between the first position f and second position s , moves the successors of the activity at the first position together to make a part <i>leftPart</i> , and moves the predecessor of the activity at the second position together to make a part <i>rightPart</i> , and then exchange the activities <i>leftPart</i> with the activities <i>rightPart</i> in the PO

The purpose of moving operations from step 6 to step 11 in the Decoding procedure is to generating a partial order which may be consistent with the given Extra Relation Representation. When moving an activity or exchanging activities in the partial order, the procedure uses link matrix as the constraints to avoid violating the finish-start constraints. Considering the extra relation defined, if a given partial order is feasible, after the operations from step 6 to step 11, the result partial order is feasible too.

Decoding supposes when an extra relation $i \rightarrow j$ between the activities i and j is used to represent a solution, the corresponding feasible solution given by Decoding procedure should have a tender that the activity i should be scheduled before the activity j . Clearly, the result feasible solution may violate the constraints decided by extra relations.

C. Selecting experimental factors

As extra relations in Extra Relation Space are symmetrical, selects relations like (i,j) as the experimental factor in which the activities i and j fulfill (4), there are two levels for every factor (i,j) , if $i \rightarrow j$ is decided, the factor has the value 0 and if $j \rightarrow i$ decided, the factor value is 1. As for the number of the factors, bigger number means more solutions generating and more moving operations in Decoding procedure.

It should be noted that for an arbitrary activities i and j , there are three link conditions between them, they are $i \rightarrow j$, $j \rightarrow i$ and $i \parallel j$. As using only conjunctions in ERR, the parallel relation $i \parallel j$ is ignored.

D. Generating Orthogonal ERRs

Given m selecting activities pairs, an orthogonal table with m factors and every factor with two levels is created. For every orthogonal array in the table, create an ERR by the meaning mentioned in above section.

E. Learning and Sampling a population

Given a population of solutions, in order to generate the next population, it is wished to get some information so that some new solutions with the history information can be gained. EDA is one of the algorithms which can achieve this purpose. In this paper, the linking frequency of an extra relation in the given population is used as the probability of this extra relation and supposes that the joint probability distribution of all the extra relations is factorized as a product of independent univariate marginal distribution (UMDA [7] p.151). Call an extra relation which is included by all the solution in the population a *full linking relation*; Call the other relations *partial linking relations*. The learning and sampling procedure is as follows:

Let $pop = \{s_i\}$ be the current population of solutions, the element s_i will include the information of start time of all the activities, makespan, fitness, and the Extra Relation Representation for it; lr , $0 \leq lr \leq 1$ be the learning rate used to select *full linking relations*; sn be the number of factors used for sampling operation; fn be the number of experimental factors used to create orthogonal table; $R(S)$ be a function to get a force of a set S . PO, P and L are defined same as subsection B.

TABLE 3. ALGORITHM.LEARNANDSAMEPLESLIST

Algorithm.LearnAndSampleSLIST($pop, lr, sn, fn, PO, P, L$)	
1.	Set $FR := \text{GetMainFactor}(pop, fn, P)$. $returnFR$
2.	Set $SM := \text{GetMainFactor}(pop, fn, P)$. $sameLink$
3.	Set $DF := \text{GetMainFactor}(pop, fn, P)$. $difLink$
4.	Set $ln := R(SM) * lr$
5.	Select ln extra relations from SM , store the selected extra relations in the set $tempLink$
6.	Set $outXList := \text{GeneratingOrthogonalERRs}(FR, P)$
7.	Add the extra relation in $tempLink$ to the project as temporary precedence constraints
8.	Set $k := 1$; $outputSList := \{\}$
9.	Repeat
10.	Get the k -th coordinate of $outXList$, $X := \{x_j\}$
11.	Random select sn extra relations from DF , the bigger frequency of an extra relation has, the bigger probability of which is selected, Set $rsnList$ is the result set.
12.	Using every extra relation in the $rsnList$ to update the coordinate X so

that the result coordinate X is corresponded with the extra relations in the $rsnList$.

13. Set $newS := \text{Decoding}(X', PO, P, L)$
 14. Add $newS$ to $outputSList$
 15. Set $k := k + 1$
 16. **Until** $k = R(outXList)$
 17. Remove the temporary precedence constraints decided by $tempLink$ from the project
 18. **Return** $outputSList$
 19. **End LearnAndSampleSLIST**
- Algorithm.GetMainFactor(pop, fn, P)**
1. For every extra relation dimension, compute the linking frequency in the input population pop , get the *full linking relations* stored in a variable $sameLink$ and *partial linking relations*. For every extra relation in the *partial linking relations*, compute the standard variance of fitness in the pop , and then select fn extra relations with the biggest standard deviation as the main factors. Use a variable $returnFR$ to store the main factors and a variable $difLink$ to store the extra relations which are not included by the $sameLink$ and $difLink$
 2. Return the variable $sameLink$, $returnFR$ and $difLink$

The *LearnAndSampleSLIST* procedure, first, at step 1, gets some extra relations by calling *GetMainFactor* procedure as the main factor, and then at step 6 gets some orthogonal ERRs by calling *GeneratingOrthogonalERRs* procedure, in order to get some different solutions, from step 4 to step 5, selects only a part of linking relations and at step 7, adds these extra relations to the project as temporary precedence constraints; from step 9 to step 16, add some random extra relations to every ERR given by the *GeneratingOrthogonalERRs* procedure, the bigger frequency of an extra relation has, the bigger probability of selecting is; at step 17, removes the temporary constraints from the project. When *LearnAndSampleSLIST* finished, some solutions which take the information of the input population and somehow are more representative, will be created.

F. Local Move

In order to improve the fitness of a given solution, the moving method provided by [14] is employed. In [14], in order to preserve precedence relations, an enhanced move of activity j at the position p_j to the left (right) forced all its predecessors (successors) to move to the left (right), while activities that was not in any precedence relation with j jumped to the other side of j . When a new partial order was gained, SSGS was used to get a feasible solution.

G. Linking two direction solutions

Given two solutions with different direction, select some activities from the first one with the increasing start time and select others activities from the second solution with the decreasing start time, regard the result partial as an AL representation, use heuristic generation scheme PSGS to create a new solution.

The reason using this method is that for a solution, generally the resource consuming rate in the front will bigger than in the back and linking some activities selected from the head of a forward direction solution with others selected from the tail of a backward direction solution may get a better solution than the two original ones.

H. Reverse Schedule

Inspired by the [15], reversing a partial order of a solution may improve this solution. In this paper, a reverse problem is constructed by reversing all the precedence constraints of original problem and keeping the resource constraints unchanged. It can be seen easily, if a partial order is feasible for the original problem, its reverse partial order is feasible for the reverse problem too. I regard the reverse problem as a coequal problem with the original one and any algorithm applied to the original problem can be applied to the reverse problem too.

Given a solution of one direction, the reverse schedule procedure will, first, get a reverse partial order of the solution, then, in the other direction problem, schedule this new partial order to gain a new solution, this new solution will have the different direction with the original one.

I Initializing populations

In order to get some solutions different with each other by the meaning of ERR, the Initializing procedure will, first, for the each direction problem, get some difference partial schedules by procedure of the branch-and-bound given by [10], and then for every partial schedule, generate some solutions randomly, at last, reverse schedule the current getting population for every direction problem, and add the new solutions to the corresponding direction population.

J. Stop criteria

In table 1, call operations from step 2 to step 20, a cycle, use the minimal makespan of the current found solution as the up bound of the problem. If the algorithm runs after some numbers of cycles, the up bound has not update, stop the algorithm and return the best found solution as the result.

IV. SIMULATION RESULTS

In order to show the efficiency of this algorithm, I simulate some bench mark instances in J30 and J60 set [16]. As some instances in these set are very simple to get an optimum solution, to simulate them will not show the efficiency of an algorithm. I select 47 problems in the J30, of which the computing time to get an optimum solution is more than 10 seconds by the best branch-and bound method and 122 problems in the J60 which have not been gained an optimum solution so far. The simulation results will be compared with the results given by [14] in which a very competitive algorithm is given. The simulation results listed in the following table will use the deviations from the best-known solution as a computing standard.

TABLE 4. SIMULATION RESULTS

Algorithm		best.dev. %		solutions generated	
		avg.	max.	avg.	max.
J30	VNS+Enh+Aug.[14]	0.01	1.56	*	*
	OrthHybirdAlg	0.3	3.4	26975	57491
J60	VNS+Enh+Aug.[14]	0.14	3.28	152503	1653641
	OrthHybirdAlg	1.6	5.1	60380	106389

Though this algorithm has not got the solution as good as [14], the number of generated solutions in this paper is smaller than [14], otherwise, I select the most difficult problems to simulate, that can be seen in above table, the maximum deviations of the two algorithm are very near, in fact, as for the 47 problems in J30, I have got all the optimum solutions for every instance by running the algorithm three times.

From the simulation results, it can be seen the algorithm given by this paper is effective.

V. CONCLUSION

A new representation for the solution of RCPSP has been provided. And the search space and distance have been defined also. By using this new representation, the linking information between arbitrary activities pair can be gain easily. That is to say the new representation provides a foundation for the linking information extracting and translating. It pays a new way to design algorithms in which the linking information is used as a basic factor to be controlled. Using the ideal of orthogonal design is only one of these algorithms. Furthermore, taking the reverse problem as a coequal problem with the original one is a new experiment in algorithm designing.

The fundamental ideal of this algorithm is that how to extract and transform information is more vital than which algorithm scheme is to be selected for algorithm designing. In fact, this ideal is the genuine meaning of the word 'heuristic'. Of course, I can not say I have found the best representation and the best algorithm, but I do believe that taking this fundamental ideal in mind will design a more effective algorithm.

REFERENCES

- [1] R.Kolisch,S.Hartmann. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. European Journal of Operational Research 127 (2000) 394-407
- [2] R.Kolisch,S.Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling : An update, European Journal of Operational Research 174 (2006) 23-37
- [3] J.Weglarz(Eds.). Project Scheduling: recent models, algorithms and applications. Dordrecht:Kluwer,1999
- [4] R.Kolisch, S.Hartmann. Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. In [3], 147-178
- [5] M.Laguna , R.Martí. Scatter Search:Methodology and implementations in C. Dordrecht:Kluwer Academic Publishers, 2003
- [6] D.Dieter, D.R.Bert, L.Roel etc.A hybrid scatter search /electromagnetism meta-heuristic for project scheduling. European Journal of Operational Research 169 (2006) 638-653
- [7] P.Larrañaga,J.A. Lozano ,Estimation of Distribution Algorithms :A new tool for evolutionary computation. Kluwer Academic Publishers,2001
- [8] J.M. Peña, V. Robles, P. Larrañaga, et al. GA-EDA: Hybrid Evolutionary Algorithm Using Genetic and Estimation of Distribution Algorithms. R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029(2004), 361-371
- [9] P.Brucker, S.Knust. Solving large-sized resource-constrained project scheduling problems.in[3], 27-51,.
- [10] E.Demeulemeester, W.Herroelen. A branch -and -bound procedure for the multiple resource-constrained project scheduling problem. Management science, 38(1992):1803-1818
- [11] Z.Huang. A new representation for the solution of resource project scheduling. Computer application, Vol.27, No.1 (2007), 222-224

- [12] Q. Wu, "On the optimality of orthogonal experimental design," *Acta Mathematicae Applicatae Sinica*, vol. 1, no. 4(1978),283–299.
- [13] Q. Zhang and Y.W.Leung. Orthogonal Genetic Algorithm for Multimedia Multicast Routing. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*. Vol.3,No.1(1999): 53-62.
- [14] K.Fleszar and K. S. Hindi. Solving the resource-constrained project scheduling problem by a variable neighbourhood search. *European journal of operational research* 155 (2004): 402—413.
- [15] R.Klein. Bidirectional planning improving priority rule-based heuristics for scheduling resource-constrained projects. *European Journal of Operational Research*,2000, 127: 619-638
- [16] <http://129.187.106.231/psplib/data.html>