Revista Colombiana de Estadística
July 2016, Volume 39, Issue 2, pp. 267 to 280
DOI: http://dx.doi.org/10.15446/rce.v39n2.51584

A Multi-Agent Proposal for the Resolution of BIBD Instances

Una propuesta multi-agente en la resolución de instancias del BIBD

David Rodríguez[1,a], Enrique Darghan[2,b], Julio Monroy[3,c]

[1]Departamento de Ing. Informática, Universidad Nacional Experimental del Táchira, San Cristóbal, Venezuela

[2]Departamento de Agronomía, Facultad de Ciencias Agrarias, Universidad Nacional de Colombia, Bogotá, Colombia

[3]Departamento de Matemáticas, Facultad de Ciencias Básicas, Universidad de Pamplona, Pamplona, Colombia

Abstract

The problem with designing balanced incomplete blocks (BIBD) is enclosed within the combinatorial optimization approach that has been extensively used in experimental design. The present proposal addresses this problem by using local search techniques known as *Hill Climbing*, *Tabu Search*, and an approach based considerable sized the use of *Multi-Agents*, which allows the exploration of diverse areas of search spaces. Furthermore, the use of a vector vision for the consideration associated with vicinity is presented. The experimental results prove the advantage of this technique compared to other proposals that are reported in the current literature.

***Key words*:** Balanced incomplete block design, Vector process, Computer search, Experimental design.

Resumen

El problema del diseño de bloques incompletos equilibrados (BIBD) se enmarca dentro del enfoque de optimización combinatoria que ha sido utilizado ampliamente en el diseño de experimentos. La presente propuesta aborda este problema utilizando técnicas de búsqueda local conocidas como *Ascenso a la Colina* (Hill Climbing), *Búsqueda Tabú* (Tabu Search) y un enfoque basado en el uso de *Multi-Agentes* que permiten la exploración de diversas áreas de espacios de búsqueda de tamaño considerable, además se presenta el uso de una visión vectorial para la consideración asociada a la

[a]Associate Professor. E-mail: drodri@unet.edu.ve
[b]Assistant Professor. E-mail: aqedarghanco@unal.edu.co
[c]Associate Professor. E-mail: julio.monroy@unipamplona.edu.co

vecindad. Los resultados experimentales evidencian la ventaja de esta técnica frente a otras propuestas mostradas en la literatura actual.

***Palabras clave*:** diseño de bloques incompletos equilibrados, proceso vectorial, búsqueda por computador, diseño de experimentos.

1. Introduction

According to Hinkelman & Kempthorne (1994), experimental designs are found to be classified in hierarchic order, depending on the number of local control or block factors. When the designs have parameters that lack precision, they are said to be Random Blocks Designs, which, depending on some inherent properties, can be identified as: Completely Randomized (CRB), Generalized Randomized (GRB), and Incompletely Randomized (IRB). The last two are characterized because not all the treatments can be applied in each block. Within the field of IRB, it is worth pointing out the design of Incomplete Balanced Blocks, which were introduced by (Yates 1936). On many occasions, experiments might require a reduction in the size of the blocks. The design of complete blocks can reduce the estimated variance of the experimental error; however this reduction might be insufficient because the number of treatments can be very large and the reduction can result impractical.

The design of incomplete blocks can avoid the complete replication of all the experimental treatments. The designs of blocks in general, are arrangements designed with the objective of having systematic control over the variability of the data attributable to external sources and to separate, minimize and even eliminate this variation from the rest of the effects of the factors of interest (Fisher 1926). The design of incomplete blocks has r of t treatments in b blocks of k experimental units with $k < t$. Raghavarao (1988), defines a BIBD as an arrangement of v symbols and b groups in which each one of the $k < v$ symbols must satisfy the following conditions: i) Each group has exactly k different symbols, ii) each symbol is present in exactly r groups, and iii) each pair of symbols are together in exactly λ groups. The parameters of a BIBD must be positive whole integers and can be identified as (v, b, r, k, λ), where v represents the number of blocks, b the number of varieties in the design, r the number of replications in each variety, k the number of different elements in each block, and λ the number of groups in which each pair of symbols occur. This must satisfy the $\lambda(v-1) = r(k-1)$ relation. This kind of technique was initially applied in experimental design (van Lint & Wilson 1992, Mead 1993): however, recently it has been applied in fields such as cryptography (Buratti 1999), codification theory (Lan, Tai, Lin, Memari & Honary 2008), sports league planning (Anderson 1997), and others.

The development of block designs is enclosed within the NP-Hard type of problems (Corneil & Mathon 1978), and it provides an excellent reference point. This is considering that it can scale and it has a wide variety of scenarios with countless instances that can be ranked according to the degree of difficulty to generate a solution. A complete number of algorithms have been applied to the problem, which continues to be unsolvable, even for designs that are relatively small in size

(Gibbons & Ostergard 2007). This article approaches the BIBD problem from a vector spaces stand point and the relationship these vectors have with each other because of their separation angle. In addition a metaheuristic technique is employed based on local search (specifically Hill Climbing) providing a resolution algorithm of great magnitude compared to other specialized algorithms found in the current literature. The BIBD problem has been addressed in recent literature by using a great number of different techniques. Some of the results obtained with these techniques can be found in Colbourn & Dinitz (1996). Much of the previous research has addressed the problem from several points of view. Specifically mathematical programming (Whitaker, Triggs & John 1990, John, Whitaker & Triggs 1993), restriction programming (Flener, Frisch, Hnich, Kzltan, Miguel & Walsh 2001, Puget 2002, Meseguer & Torras 2001). Regarding restriction programming, it is convenient to mention that a point of interest for researchers has been the symmetric property (permutation of rows and columns that can produce the same results) present in the problem. In term of this problem, Puget (2002) proposes a combination of methods to solve the problem through the rupture of symmetries, while Meseguer & Torras (2001) explore two strategies (a heuristic one for the selection of variables and a trimming procedure).

Recent proposals have used heuristic techniques to resolve BIBD. For this reason, the generation of the BIBD type designs was formulated by Bofill, Guimerà & Torras (2003) as a combinatory optimization problem addressed at neural networks. The simulated annealing algorithm that is supplied with a neural network (NN-SA) has proven to offer a better performance than the analogous hybridizations with simulated annealing. These results were improved by Prestwich (2003*a*, 2003*b*), using an local search algorithm in conjunction with the addition of new restrictions that were a result of with the symmetric outline of the problem. Rodriguez, Cotta & Fernandez (2009), recently improved the results obtained by the NN-SA and CLS algorithms. They considered the use of two local search techniques (hill climbing, tabu search), as well as a technique based on genetic algorithms. To do this, two forms of vicinity were used, the change of ones for zeros (bit-flip) and the exchange of pairs (0,1) or (1,0) (position-swapping); results reveal the superiority of the local search technique: the Tabu Search with swap vicinity. The same authors presented an approach based on the application of a hybrid nature memetic algorithm (Rodriguez, Cotta & Leiva 2011), in which was extended the number of evaluations of the target function to 2×10^7 and a new crossing operator was incorporated named *Greedy*. A total of 63 of 86 instances was solved with this approach. Finally, Daisuke Yokoya (2009), propose a mathematical model based on the restrictions that a BIBD must meet, the local search technique Tabu Search, and the resolver of linear models CPLEX, in order to provide a solution to the well-known designing balanced incomplete blocks problem. This proposal shows the best results obtained to date; a total of 78 of the 86 instances studied in the proposal were solved.

2. Methods

Using the notation $\langle v, b, r, k, \lambda\rangle$, a $\langle v, b, r, k, \lambda\rangle$-BIBD problem consists of the grouping of a series of v objects in b groups of $k < v$ objects each in such a way that a specific object in v belongs to r different groups, and any pair of objects in v will be present in exactly $\lambda < b$ groups. A standard form of representation for a solution to the problem could be expressed in terms of the incidence matrix, $M \equiv \{m_{ij}\}_{v\times b}$, as is presented in Hall (1998). This is subject to the following:

$$m_{ij} = \begin{cases} 1 & 1 \text{ if treatment } i \text{ appears in block } j \\ 0 & 1 \text{ if treatment } j \text{ does not appears in block } j. \end{cases}$$

Furthermore, $\sum\limits_{j=1}^{b} m_{ij} = r$, $\sum\limits_{i=1}^{v} m_{ij} = k$, $\sum\limits_{j=1}^{b} m_{ij}m_{i'j} = \lambda$; $i = 1, 2, \cdots, v$; $j = 1, 2, \ldots, b$; with $i \neq i'$;$i, i' = 1, 2, \ldots, v$. An example of this representation can be observed in Figure 1, which shows the incidence matrix M for a possible solution to the instance $\langle 8, 14, 7, 4, 3\rangle$-BIBD.

$$\begin{matrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{matrix}$$

Figure 1: Candidate solution for the instance $\langle 8, 14, 7, 4, 3\rangle$-BIBD.

The relation between the parameters $\langle v, b, r, k, \lambda\rangle$ must satisfy $vr = kb$; $r(k-1) = \lambda(v-1)$; $\lambda = \frac{r(k-1)}{(v-1)}$ and $b > v$ (Fisher's (1940) inequality). Also, if $v = b$ and $r = k$, we have a symmetric design.

2.1. Vectorial Approach for $\langle v, b, r, k, \lambda\rangle$-BIBD

If we analyze the incidence matrix, we can see that the representation of each one of the rows of the matrix is given by a binary components vector in the space R^n. The totality of the vector space represents the possible v treatments of the design to be built. Evidently not all the vectors correspond to feasible treatments because some of them will not meet the necessary conditions that a valid BIBD must have. On the other hand, the condition $\sum_{j=1}^{b} m_{ij}m_{i'j} = \lambda$ allows a direct relationship to be established between the value of λ and the α angle between two specific vectors of the vector space. According to vector algebra we have that $u, \nu \in R^b$, with binary components, and with the operations of the scalar product of vectors, vector module, inequality of Cauchy-Schwarz, and the angle between two vectors, allows us to establish that, for any pair of vectors m_{1j} y m_{2j}, with $j = 1, 2, \ldots, b$, the following must be met:

$$m_{1j}m_{2j} = \lambda \tag{1}$$

This represents a necessary and sufficient condition for a specific vector from the vector space to be part of a feasible treatment that will allow the construction of a valid BIBD. Therefore, we deduce that $u.v = \lambda$. In addition, $\sum_{j=1}^{b} u_j^2 = r$, considering that $u_j \in \{0, 1\}$. Because of this, $|u| = |\nu| = \sqrt{r}$, from which it results that the α angle formed by two vectors that represent a feasible treatment, is given by $\alpha = \arccos\left(\frac{\lambda}{r}\right)$. Additionally to this condition, the parameter k should be considered to determine the number of objects in each group b. This will allow the establishment of another condition that a specific vector present in the vector space should meet in order to be considered a feasible treatment and to establish a valid BIBD. Thus, the total number of elements in the vector space is given by $\binom{b}{r}$, and if the condition $\sum_{i=1}^{v} m_{ij} = k$ is introduced, only a subspace of the vector space is used.

2.2. Local Search Technique for $\langle v, b, r, k, \lambda \rangle$-BIBD

This section describes the proposal for the utilization of the metaheuristic techniques Hill Climbing and Tabu Search. It taken into consideration the considered vicinity, the multi-agent proposal, as well as the objective function and the elements of the local search.

2.2.1. Neighborhood

When processing a feasible treatment, for example, considering vector u as a starting point for the construction of a BIBD, whether v is a valid candidate treatment should be verified. For the instance: $\langle 14, 26, 13, 7, 6 \rangle$, with

$$u = \langle 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1 \rangle,$$

and considering that $\nu = \langle 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1 \rangle$, then $\alpha = \arccos\left(\frac{\lambda}{r}\right) = \arccos\left(\frac{6}{13}\right) = 1.0911$, with $u.\nu = 6$, $|u| = \sqrt{13}$ and $|u||\nu| = 13$. As such, vector v is a valid candidate treatment. With another vector, for example $\nu' = \langle 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0 \rangle$, you can verify that the angle between $\langle u, \nu' \rangle$ or $\langle \nu, \nu' \rangle$ is different because $1.0911 \neq 1.0697$, which indicates that ν' does not belong to the considered vicinity. Another important aspect has to do with the reduction of the vicinity as well as guiding the search; this relates to the condition $\sum_{i=1}^{v} m_{ij} = k$. Continuing with the previous example, after several iterations, you obtain a candidate BIBD,BI_7 equal to:

$$BI_7 = \begin{pmatrix} 0001101111110010011000011 \\ 11011010100001011001001011 \\ 01101001010011010111000101 \\ 10000100110111111110001000 \\ 00111010100011100010111100 \\ 00000111101001011100110101 \\ 10100111000010101011100011 \end{pmatrix}$$

According to the condition, to following is obtained:

$$32334354532245444453323335$$

If the k parameter is considered, the number of some of the missing in each column would be:

$$45443423245532333324454442$$

Through this condition we can see, that it is possible to try to guide the process primarily considering search those vectors that meet this restriction. The proposal has considered prioritizing the evaluation of those vectors of the vector space with components of the incidence matrix, with values taken from the first λ-columns where there is a higher deficit of ones, besides the angle between the vectors that already conforms the candidate BIBD. As such, a new candidate could be

$$01001111001110110001011000$$

After evaluating the angle conditions and the condition $\sum_{j=1}^{b} m_{ij}m_{i'j} = \lambda$, the new condition will be considered part of the solution and, because of this, it will be incorporated into the BI candidate.

$$BI_8 = \begin{pmatrix} 0001101111110010011000011 \\ 11011010100001011001001011 \\ 01101001010011010111000101 \\ 10000100110111111110001000 \\ 00111010100011100010111100 \\ 00000111101001011100110101 \\ 10100111000010101011100011 \\ 01001111001110110001011000 \end{pmatrix}$$

However, if the priority be vectors to considered are not feasible treatments, you proceed to take the components according to the lowest number of violations of the restrictions.

There is a marked difference between our approach and the are presented by Rodriguez et al. (2011). The first one uses a vectorial scheme that allows major levels of exploration (i.e diversification) due to the fact that the measure of distance used here to separate the neighbors is different from that used in Rodríguez. This allow zones of search space that are more distant to be explored.

2.2.2. Objective Function

The objective function is expressed throughout, $\sum_{j=1}^{b} m_{ij}m_{i'j} = \lambda$ and:

$$f^{\langle v,b,r,k,\lambda\rangle}(M) = \sum_{i=1}^{v} \phi_{ir}(M) + \sum_{j=1}^{b} \phi'_{jk}(M) + \sum_{i=1}^{v-1} \sum_{j=i+1}^{v} \phi''_{ij\lambda}(M) \qquad (2)$$

with $\phi_{ir}(M) = \left| r - \sum_{j=1}^{b} m_{ij} \right|$, $\phi'_{jk}(M) = \left| k - \sum_{i=1}^{v} m_{ij} \right|$,
$\phi''_{ij\lambda}(M) = \left| \lambda - \sum_{k=1}^{b} m_{ik}m_{jk} \right|$

Note that 2 will only be applied when the totality of the treatments necessary to construct the desired BIBD is available. As long as $\sum_{j=1}^{b} m_{ij} = r$, it should be applied each time there is a feasible treatment to incorporate. If a BIBD candidate contains v treatments, the global optimum will be reached if there is a configuration M^* so that $f^{\langle v,b,r,k,\lambda\rangle}(M^*) = 0$.

2.2.3. Local Search Techniques

In this proposal, the metaheuristic techniques of Hill Climbing (H_c) and Tabu Search (T_s) have been used in the search for a valid BIBD. Basically, the process consists of taking a candidate treatment ν_i from the vector space $\varphi \in R^b$, so that:

$$\{\nu_i \in \varphi \mid \nu_i \in \binom{b}{r} i = 1, 2, \ldots, v\}$$

where $\langle v, b, r\rangle$ represents parameters in the design that is being locked for. If ν_i satisfies the angular conditions and the $\sum_{j=1}^{b} m_{ij}m_{i'j} = \lambda$, it should be included as a BIBD candidate BI_i. And, if any of the restrictions are not satisfied, the vector is discarded and a new vector is taken from φ. It is clear that the number of neighbors available to be evaluated is a considerable size, mainly for instances with a greater number of blocks. This makes it difficult to evaluate the totality of the vector space. Because of this, for example, for an instance like $\langle 14, 26, 13, 7, 6\rangle$, there is a total of $6.476475253248 \times 10^{16}$ neighbors to process. For the case of the H_c technique, after the evaluation of b^2 vectors from the vector space without being able to add a new treatment, the search is restarted from another starting point (new vector generated randomly from R^b that will meet the restriction in the incidence matrix where the value of the parameter r will indicate the amount of components with value "one" and the remainders with value "zero"). Clearly BI will contain, once again, only one vector $i = 1$ (BI_1). Additionally, the tabu search algorithm has been considered and is, also defined on the basis of the vector space for its vicinities, with an exploration equal to H_c. Other parameters important to the Tabu algorithm are: a) a vector $\nu_i \in \varphi$ previously evaluated is considered "tabu", b) the tabu persistence is chosen based on the parameter v from the instance that will be processed, c) the desired criteria consists in finding a new feasible vector that will be a valid treatment, d) after the exploration of b^2 vectors from the vector space, the search is intensified or diversified. The intensification is undertaken by going back over the best solution that can be found by substituting one of the vectors from the BIBD under construction (BI_{i-1}). This vector will be selected randomly from the previously placed i vectors. Diversification is undertaken by restarting from a new vector of the vector space φ, as long as it meets $\sum_{j=1}^{b} m_{ij} = r$, with a probability of $1/16$.

Besides the techniques described in Section 2.2.3, we have also introduced two Multi-Agent models based on: (a) the studied metaheuristics (b) our approach and the approach presented by Daisuke Yokoya (2009). Both cases uses a centralized outline with an independent search has been used.

2.2.4. Multi-Agent Proposal: Hc and Ts Agents

This approach only includes the metaheuristics presented at the beginning of this section and a group of 30 individuals initial solutions are applied, which will later be delivered to the cooperating algorithms. The group of algorithms employed corresponds to the two proposals presented in this article H_c and T_s. Other relevant parameters used can consider one of the proposals for each of the two versions: a version that randomly takes a vector from the vector space as an initial candidate a version that takes a possible design as an initial candidate (surely with various vectors from the space that have been proven as valid treatments) from the group of feasible candidates. Additionally, the group of candidate solutions will begin with the application of the algorithm T_s. It will have a random start for approximately 1 second of CPU time, and a total of 30 runs. At one moment of time there will only be two algorithms running simultaneously. After two consecutive runs of a specific algorithm without contribution, it will be replaced by another algorithm from the group. It is said that there is no contribution if the solution obtained by the algorithm is worse than the existing ones, or if the obtained individual is the same as one of the existing ones. The method of assigning an initial solution can be done in several ways: a) take a solution randomly from the group as long as it is not tabu, b) assign according to a quality percentage of the individuals, without considering the tabu condition, and c) assign the best individual of the group without considering the tabu condition. The quality percentage applied has been 25%. A higher percentage was tried but it did not show improvement. The tabu persistence used was based on the value of the parameter v for each instance. The cooperation was executed until a solution was obtained or a maximum of 30 runs was reached. Finally, each run was performed for a maximum of approximately 500 seconds of the CPU.

2.2.5. The $TABU(20)$, Hc and Ts Agents

In this scenario, besides the algorithms Hc and Ts a proposal named $TABU(20)$ presented by Daisuke and Takeo has been included. The mechanism employed is similar to the one presented in section 2.2.4, with slight differences, such as: a) the group of algorithms is composed by three algorithms that correspond to H_c, T_s and $TABU(20)$ in their cooperating version. The non-cooperating version of

T_s has been used to initiate the group of initial candidates, b) a group of 100 initial solutions has been employed, c) for the initiation of the group algorithm, T_s has been used, d) cooperation was executed until a solution was obtained or a maximum of 25 runs was reached, and e) each run was performed for a maximum of approximately 100 seconds of the CPU.

3. Results

The tests were performed on 86 instances taken from Prestwich (2003*a*), Rodriguez et al. (2009), Bofill et al. (2003) with $vb \leq 1000$ and $k \neq 3$, see table 1. These correspond to the cases that have a higher degree of complexity to find the solution reported in these works. It was taken into account that for those instances where $k = 3$ it is considerably less complicated to find a solution.

TABLE 1: Group of instances considered in the experimentation.

ID	v	b	r	k	λ	vb	ID	v	b	r	k	λ	vb	ID	v	b	r	k	λ	vb
1	8	14	7	4	3	112	30	9	54	24	4	9	486	59	21	35	15	9	6	735
2	11	11	5	5	2	121	31	13	39	12	4	3	507	60	10	75	30	4	10	750
3	10	15	6	4	2	150	32	13	39	15	5	5	507	61	25	30	6	5	1	750
4	9	18	8	4	3	162	33	16	32	12	6	4	512	62	20	38	19	10	9	760
5	13	13	4	4	1	169	34	15	35	14	6	5	525	63	16	48	15	5	4	768
6	10	18	9	5	4	180	35	12	44	22	6	10	528	64	16	48	18	6	6	768
7	8	28	14	4	6	224	36	23	23	11	11	5	529	65	12	66	22	4	6	792
8	15	15	7	7	3	225	37	10	54	27	5	12	540	66	12	66	33	6	15	792
9	11	22	10	5	4	242	38	8	70	35	4	15	560	67	9	90	40	4	15	810
10	16	16	6	6	2	256	39	17	34	16	8	7	578	68	13	65	20	4	5	845
11	12	22	11	6	5	264	40	10	60	24	4	8	600	69	11	77	35	5	14	847
12	10	30	12	4	4	300	41	11	55	20	4	6	605	70	21	42	10	5	2	882
13	16	20	5	4	1	320	42	11	55	25	5	10	605	71	21	42	12	6	3	882
14	9	36	16	4	6	324	43	18	34	17	9	8	612	72	21	42	20	10	9	882
15	8	42	21	4	9	336	44	25	25	9	9	3	625	73	16	56	21	6	7	896
16	13	26	8	4	2	338	45	15	42	14	5	4	630	74	10	90	36	4	12	900
17	13	26	12	6	5	338	46	21	30	10	7	3	630	75	15	60	28	7	12	900
18	10	36	18	5	8	360	47	16	40	10	4	2	640	76	18	51	17	6	5	918
19	19	19	9	9	4	361	48	16	40	15	6	5	640	77	22	42	21	11	10	924
20	11	33	15	5	6	363	49	9	72	32	4	12	648	78	15	63	21	5	6	945
21	14	26	13	7	6	364	50	15	45	21	7	9	675	79	16	60	15	4	3	960
22	16	24	9	6	3	384	51	13	52	16	4	4	676	80	16	60	30	8	14	960
23	12	33	11	4	3	396	52	13	52	24	6	10	676	81	31	31	6	6	1	961
24	21	21	5	5	1	441	53	10	72	36	5	16	720	82	31	31	10	10	3	961
25	8	56	28	4	12	448	54	19	38	18	9	8	722	83	31	31	15	15	7	961
26	10	45	18	4	6	450	55	11	66	30	5	12	726	84	11	88	40	5	16	968
27	15	30	14	7	6	450	56	22	33	12	8	4	726	85	22	44	14	7	4	968
28	16	30	15	8	7	480	57	15	52	26	7	12	780	86	25	40	16	10	6	1000
29	11	44	20	5	8	484	58	27	27	13	13	6	729							

The tests were executed on an Intel computer with a dual core processor at 2.5 GHZ. Three different scenarios have been presented for the tests. These correspond with the local searches, the cooperation between the two proposed local searches, and the cooperation between the local searches and the technique described by Daisuke y Takeo. The maximum time used in each run for the local searches was approximately 500 seconds: the same time was used in the cooperation proposal that only involves the local searches. In the second cooperation proposal, each run was executed in a maximum of 100 seconds of the CPU. Results present the best times obtained to be resolved each one of the indicated instances. Table 2 presents the different results obtained in the recent research. The first column corresponds to the identification number of each instance, the second column shows the results obtained by Prestwich (2003a), the third column submarines the results presented by Rodriguez et al. (2009), and finally, columns 4 and 5 illustrate the results obtained by Daisuke Yokoya (2009).

Table 2: Recent results for the 86 instances. Each column represents the different proposals presented and the time (seconds) used to reach the solution.

ID	CLS	T_{ss}	BAB	$TABU$	ID	CLS	T_{ss}	BAB	$TABU$	ID	CLS	T_{ss}	BAB	$TABU$
1	0.00	Si	0.00	0.00	30	0.21	Si	0.01	0.00	59	–	–	–	–
2	0.04	Si	0.01	0.01	31	1.22	Si	0.03	0.02	60	0.92	Si	0.02	0.02
3	0.04	Si	2068.03	0.02	32	11.20	Si	0.05	0.01	61	14.30	Si	0.04	0.03
4	0.05	Si	0.01	0.00	33	–	–	1985.33	3.57	62	–	–	–	–
5	0.01	Si	0.00	0.00	34	–	–	3.29	0.05	63	43.20	Si	0.11	0.15
6	0.12	Si	0.01	0.01	35	27.90	Si	0.04	0.03	64	–	–	0.38	0.37
7	0.06	Si	0.00	0.00	36	–	Si	–	0.07	65	1.38	Si	0.05	0.02
8	0.61	Si	0.01	0.01	37	0.98	Si	0.04	0.02	66	12.80	Si	0.04	0.03
9	1.20	Si	0.01	0.01	38	0.13	Si	0.01	0.00	67	0.69	Si	0.02	0.02
10	0.54	Si	–	0.02	39	–	–	–	7.37	68	1.40	Si	0.04	0.01
11	1.23	Si	0.02	0.02	40	0.75	Si	0.01	0.01	69	5.11	Si	0.03	0.02
12	0.12	Si	0.05	0.01	41	0.80	Si	0.03	0.01	70	–	–	–	798.86
13	0.16	Si	0.01	0.01	42	4.45	Si	0.02	0.06	71	–	–	–	–
14	0.21	Si	0.01	0.00	43	–	–	–	30.91	72	–	–	–	–
15	0.25	Si	0.01	0.00	44	–	–	–	0.24	73	–	–	0.12	0.11
16	0.50	Si	0.04	0.06	45	42.90	Si	0.14	0.05	74	0.99	Si	0.04	0.01
17	7.93	Si	17.54	0.08	46	–	–	–	61.42	75	–	Si	0.09	0.05
18	1.08	Si	0.03	0.03	47	4.22	Si	0.05	0.08	76	–	–	0.78	1.34
19	9.73	Si	0.21	0.05	48	–	–	2.72	0.84	77	–	–	–	–
20	1.79	Si	0.04	0.01	49	0.54	Si	0.01	0.01	78	30.70	Si	0.09	0.20
21	–	–	0.38	0.88	50	–	–	0.06	0.06	79	6.31	Si	0.05	0.02
22	9.80	Si	0.03	0.05	51	3.76	Si	0.03	0.01	80	–	–	0.34	1.12
23	0.33	Si	–	0.22	52	49.70	Si	0.04	0.02	81	2.04	Si	0.07	0.04
24	0.22	Si	0.31	0.02	53	1.33	Si	0.02	0.01	82	–	–	–	1.70
25	0.12	Si	0.01	0.01	54	–	–	–	–	83	–	–	–	0.10
26	0.08	Si	0.02	0.01	55	1.52	Si	0.04	0.01	84	4.07	Si	0.03	0.04
27	–	–	10.13	0.95	56	–	–	–	–	85	–	–	–	–
28	–	–	237.50	0.44	57	–	–	0.11	0.07	86	–	–	–	–
29	1.13	Si	0.04	0.01	58	–	–	–	0.54					

Table 3 compiles the experimental results obtained from this investigation is proposal. Columns $2-3$ present the results obtained when applying the proposed local search methods Hill Climbing (H_c) and Tabu Search (T_s). We can observe that algorithm T_s is able to resolve 72 of the 86 instances compared whit H_c, which resolves 70 of the 86. The response times in both proposals are relatively similar. In column 4 we observe the results when applying a cooperation model in which algorithms H_c and T_s are applied. In this case 74 of the 86 instances are resolved. Finally column 5 summarizes the best times obtained by the cooperative proposal, including the algorithms. There are presented here as will as in the model presented in Daisuke Yokoya (2009). Table 4 presents a summary of the number of instances resolved by the different revised algorithms that are presented in this research.

Table 3: Experimental results for the 86 instances. The best results are from the proposals H_c, T_s, cooperation $H_c + T_s$ and cooperation $TABU(10) + H_c + T_s$ $(TABU++)$. Each column represents the different proposals presented and the time (seconds) used to reach the solution.

ID	H_c	T_s	H_c+T_s	$TABU++$	ID	H_c	T_s	H_c+T_s	$TABU++$					
1	0.03	0.03	0.02	0.00	30	0.01	0.01	0.01	0.00	59	–	–	–	–
2	0.02	0.03	0.02	0.00	31	0.01	0.03	0.03	0.00	60	0.05	0.03	0.03	0.01
3	0.03	0.01	0.01	0.00	32	0.05	0.03	0.03	0.01	61	3.56	36.50	0.02	0.00
4	0.01	0.01	0.01	0.01	33	13.72	24.78	49.72	0.94	62	–	–	–	–
5	0.03	0.03	0.03	0.00	34	5.70	18.38	0.81	0.80	63	4.11	5.59	0.17	0.16
6	0.06	0.05	0.06	0.03	35	2.69	2.17	0.41	0.39	64	19.92	27.48	0.73	0.72
7	0.02	0.01	0.03	0.00	36	0.72	0.61	0.56	0.55	65	0.05	0.05	0.03	0.02
8	0.03	0.01	0.01	0.01	37	0.13	0.09	0.09	0.06	66	28.36	37.09	35.36	0.66
9	0.08	0.25	0.23	0.22	38	0.05	0.03	0.05	0.03	67	0.36	0.36	0.34	0.31
10	0.03	0.03	0.01	0.00	39	–	326.77	312.66	14.92	68	0.03	0.03	0.03	0.01
11	0.03	0.03	0.03	0.01	40	0.05	0.03	0.03	0.01	69	1.42	26.36	1.02	0.98
12	0.03	0.01	0.01	0.00	41	0.03	0.01	0.03	0.00	70	–	–	–	1.19
13	0.13	0.23	0.23	0.22	42	0.08	0.06	0.06	0.05	71	–	–	–	–
14	0.03	0.03	0.03	0.00	43	–	–	–	1.64	72	–	–	–	–
15	0.01	0.03	0.01	0.00	44	11.92	38.16	0.39	0.38	73	9.17	7.72	1.01	0.92
16	0.28	0.76	0.72	0.70	45	7.58	2.94	0.11	0.09	74	0.19	0.14	0.13	0.11
17	0.03	0.03	0.03	0.01	46	–	–	89.30	4.19	75	37.23	29.00	27.70	0.28
18	0.05	0.05	0.05	0.03	47	9.02	2.13	0.01	0.00	76	29.67	11.83	0.91	0.88
19	0.14	0.76	0.73	0.72	48	7.66	35.25	0.28	0.28	77	–	–	–	–
20	0.01	0.03	0.01	0.00	49	16.53	13.92	0.03	0.01	78	0.44	0.33	0.33	0.30
21	13.11	2.72	0.72	0.72	50	37.73	44.92	42.59	1.14	79	0.03	0.03	0.05	0.01
22	0.17	0.19	0.19	0.17	51	0.03	0.01	0.03	0.00	80	–	–	–	0.30
23	0.01	0.01	0.01	0.00	52	1.39	1.26	0.83	0.81	81	0.03	0.03	0.01	0.01
24	0.11	30.26	0.01	0.00	53	1.59	1.36	0.67	0.66	82	–	–	469.81	26.38
25	4.33	3.89	0.02	0.01	54	–	–	–	58.14	83	–	229.88	187.25	5.67
26	0.01	0.03	0.03	0.00	55	1.14	0.98	0.92	0.91	84	7.39	6.95	0.66	0.64
27	6.69	3.08	0.22	0.20	56	–	–	–	–	85	–	–	–	–
28	19.16	10.44	9.94	0.44	57	13.50	1.83	61.41	43.17	86	–	–	–	–
29	0.05	0.03	0.03	0.01	58	26.92	36.97	98.72	0.70					

Table 4: Summary of results.

Method	Resolved	Unresolved
CLS	55	31
Tabu Search Swap	57	29
BAB BIBD(LP.BWD)	64	22
BAB BIBD(LP, FWD)	66	20
TABU BIBD(10)	77	9
H_c	70	16
T_s	72	14
$H_c + T_s$	74	12
$TABU++$	78	8

4. Conclusions

We have presented an algorithm with a constructive approach which allows a great number of instances to be resolved for the problem of the design of balanced incomplete blocks using a technique of local search based on a multi-start. It can be confirmed that by using cooperative mechanisms and incorporating multiple agents you can achieve a better performance from the techniques. However, there are still instances for which it has not been possible to reach optimal solutions, but it should be mentioned that the feasible solutions found are very close to the optimal globals. However, an advantage of our approach lies in the major capacity to explore of new zones (major diversification). This is unlike other approaches that use major intensification, for example the vectorial approach. Normally one of the disadvantages of the process of intensification is the wide possibility of finding many local optima: something that can be solved by using the diversification method.

In practical situations, adoption of a complete block design is not appropriate and in several cases, not at all feasible. This fact prompted the development of various kinds of incomplete block designs, which, in turn, have been used extensively for experiments in differents areas. Moreover, these designs opened up many challenging problems in combinatorial optimization. In addition to the classical IBD (incomplete block designs), the BIB (balanced incomplete block) and PBIB (partially balanced incomplete block) designs are still found tu be useful in several applications.

We believe that the benefits of this proposal seem to be very promising. For this reason we have presented the work hypothesis that an adequate specialization of the search algorithms could open the door to resolve of many of these instances. Therefore, in terms of future work, we consider hybridizing the presented proposals, employing genetic algorithms and methods of local search. From another perspective, we can attempt to employ a different model the presentation the candidate solutions, as well as, study other local search techniques: Iterated Local Search (ILS), Variable Neighborhood Search (VNS).

[Received: September 2015 — Accepted: May 2016]

References

Anderson, I. (1997), *Combinatorial designs and tournaments*, Clarendon Press, Oxford University Press.

Bofill, P., Guimerà, R. & Torras, C. (2003), 'Comparison of simulated annealing and mean field annealing as applied to the generation of block designs', *Neural Networks* **16**(10), 1421–1428.

Buratti, M. (1999), 'Some (17q, 17, 2) and (25q, 25, 3)BIBD constructions', *Designs, Codes and Cryptography* **16**(2), 117–120.

Colbourn, C. & Dinitz, J., eds (1996), *The CRC handbook of combinatorial designs*, CRC Press, Boca Raton.

Corneil, D. G. & Mathon, R. (1978), 'Algorithmic techniques for the generation and analysis of strongly regular graphs and other combinatorial configurations', *Annals of Discrete Mathematics* **2**, 1–32.

Daisuke Yokoya, T. Y. (2009), 'A mathematical programming approach to the construction of bibds', *International Journal of Computer Mathematics* pp. 1–16.

Fisher, R. A. (1926), 'The arrangement of field experiments', *Journal of the Ministry of Agriculture Great Britain* **33**.

Fisher, R. A. (1940), 'An examination of the different possible solutions of a problem in incomplete blocks', *Annals of Eugenics* **10**, 52–75.

Flener, P., Frisch, A. M., Hnich, B., Kzltan, Z., Miguel, I. & Walsh, T. (2001), Matrix modelling, *in* 'CP-01 Workshop on Modelling and Problem Formulation. International Conference on the Principles and Practice of Constraint Programming'.

Gibbons, P. B. & Ostergard, P. R. J. (2007), Computational methods in design theory, *in* C. J. Colbourn & J. H. Dinitz, eds, 'Handbook of Combinatorial Designs', Chapman & Hall/CRC Press, Boca Raton, pp. 755–783.

Hall, M. J. (1998), *Combinatorial Theory*, 2 edn, John Wiley & Sons, Inc., New York, USA.

Hinkelman, K. & Kempthorne, O. (1994), *Design and analysis of experiments*, Vol. 1, John Wiley and Sons, Inc., New York.

John, J. A., Whitaker, D. & Triggs, C. M. (1993), 'Construction of cyclic designs using integer programming', *Journal of statistical planning and inference* **36**(2), 357–366.

Lan, L., Tai, Y. Y., Lin, S., Memari, B. & Honary, B. (2008), 'New constructions of quasi-cyclic LDPC codes based on special classes of BIDBs for the AWGN and binary erasure channels', *IEEE Transactions on Communications* **56**(1), 39–48.

Mead, R. (1993), *Design of Experiments: Statistical Principles for Practical Applications*, Cambridge University Press.

Meseguer, P. & Torras, C. (2001), 'Exploiting symmetries within constraint satisfaction search', *Artificial Intelligence* **129**(1-2), 133–163.

Prestwich, S. (2003*a*), A local search algorithm for balanced incomplete block designs, *in* F. Rossi, ed., '9th International Conference on Principles and Practices of Constraint Programming (CP2003)', LNCS, Springer, pp. 53–64.

Prestwich, S. (2003*b*), 'Negative effects of modeling techniques on search performance', *Annals of Operations Research* **18**, 137–150.

Puget, J.-F. (2002), Symmetry breaking revisited, *in* P. V. Hentenryck, ed., '8th International Conference on Principles and Practice of Constraint Programming (CP 2002)', Vol. 2470 of *LNCS*, Springer, Ithaca, NY, USA, pp. 446–461.

Raghavarao, D. (1988), *Constructions and Combinatorial Problems in Design of Experiments (Paperback)*, Dover Publications.

Rodriguez, D., Cotta, C. & Fernandez, A. (2009), Finding balanced incomplete block designs with metaheuristics, *in* C. Cotta & P. Cowling, eds, 'Evolutionary Computation in Combinatorial Optimization 2009', Vol. 5482 of *Lecture Notes in Computer Science*, Springer, pp. 156–167.

Rodriguez, D., Cotta, C. & Leiva, A. J. F. (2011), 'A memetic algorithm for designing balanced incomplete blocks', *IJCOPI* **2**(1), 14–22.

van Lint, J. & Wilson, R. (1992), *A Course in Combinatorics*, Cambridge University Press.

Whitaker, D., Triggs, C. M. & John, J. A. (1990), 'Construction of block designs using mathematical programming', *Journal of the Royal Statistical Society, Series B* **52**(3), 497–503.

Yates, F. (1936), 'Incomplete randomized blocks', *Annals of Eugenics* **7**, 121–140.