International Journal of Group Theory
ASSN (print): 2251-7650, ISSN (on-line): 2251-7669
Vol. x No. x (202x), pp. xx-xx.

http://ijgt.ui.ac.ir

www.ui.ac.ir

DETERMINATION OF THE LEFT BRACES OF ORDER 64

ADOLFO BALLESTER-BOLINCHES AND RAMÓN ESTEBAN-ROMERO* AND VICENT PÉREZ-CALABUIG

Abstract. We review some of the algorithms used for the determination of the left braces of order 64.

1. Introduction

We say that $(B, +, \cdot)$ is a *skew left brace* when B is a set and $+$ and $\cdot$ are two binary operations on B such that $(B, +)$ and $(B, \cdot)$ are groups linked by the distributive-like law $x(y+z) = xy - x + xz$ for x, y, $z \in B$. This structure was introduced by Guarnieri and Vendramin in [10] as a generalisation of the *left braces* introduced by Rump in [11], that are the skew left braces in which the operation $+$ is commutative. We will follow the convention of omitting the symbol $\cdot$ and writing $-a$ for the symmetric of $a \in B$ in $(B, +)$ and $a - b$ for $a + (-b)$. As in the usual arithmetic, multiplications are evaluated before additions.

One of the most natural problems in the study of finite skew left braces is to determine the isomorphism classes of (skew) left braces of a given order. The computation of a list of representatives of these isomorphism classes would be a powerful tool that would allow the use of computers to check the validity of some claims about left braces and, in particular, to look for counterexamples. Since skew left braces give non-degenerate set-theoretic solutions to the Yang-Baxter equation, an important

Communicated by Patrizia Longobardi.

MSC(2020): Primary: 16T25; Secondary: 81R50, 20-08

Keywords: Skew left braces, left brace, holomorph, regular subgroup.

Article Type: Ischia Group Theory 2022.

*Corresponding author.

Received: 25 May 2023, Accepted: 03 August 2023.

http://dx.doi.org/10.22108/ijgt.2023.137825.1846 .

equation of the theoretical physics (see [10, Section 3]), finding all finite skew left braces with a given order would be a step forward in the search of set-theoretic solutions of the Yang-Baxter equation.

Guarnieri and Vendramin presented in [10, Algorithm 5.1] a general algorithm (see Section 4 below) to compute all isomorphism classes of skew left braces with a given additive group A. With the help of this algorithm, they obtained the numbers of isomorphism classes of left braces of all orders up to 120 except for 32, 64, 81, and 96. These numbers appear in [10] and were obtained with a Magma [7] implementation of that algorithm. Representatives of the isomorphism classes of these skew left braces form part of the YangBaxter package [13] for GAP [9]. The determination of the isomorphism classes of skew left braces for the remaining cases was left as an open problem ([10, Problem 6.1]). Some partial results were obtained in [12, Table 2.3]. In that paper, the numbers of isomorphism classes of (skew) left braces of sizes 64, 96, or 128 were considered "extremely large" and the current "computational methods not strong enough to construct them all."

Bardakov, Neshchadim, and Yadav improved in [6, Algorithm 2.4] this algorithm to compute all isomorphism classes of finite skew left braces with a given additive group. They applied this algorithm to obtain all isomorphism classes of skew left braces of orders 32 and 81, as well as the number of isomorphism classes of left braces of order 96. They also gave the numbers of isomorphism classes of left braces whose additive group is an abelian group of order 64, except for the cases of additive groups $C_4 \times C_4 \times C_4$ (`SmallGroup(64, 55)` in the GAP [9] library for groups of small order) and $C_2 \times C_2 \times C_4 \times C_4$ (`SmallGroup(64, 192)` in this library). The authors have determined the isomorphism classes of left braces for these additive groups in [3, 5], respectively. This concludes the problem of the classification of left braces, that is, skew left braces with abelian additive group, of order 64 up to isomorphism. In this survey we present some of the ideas that we have used to determine these left braces.

Here and in [3, 5] we have focused only on skew left braces with abelian additive group, i. e., left braces. The classification of all skew left braces of order 64 with non-abelian additive group is still an open problem, but we believe that it could be solved with similar techniques and perhaps with the help of large computational resources.

2. The holomorph of a group and its regular subgroups

The results of this section, that can also be found in [3], are well known and we state them just for the sake of completeness.

Suppose that $(G,+)$ is a group. The *holomorph* $\mathrm{Hol}(G,+)$ of $(G,+)$ is the semidirect product of G with the automorphism group of G. If (g,α) and (h,β) are two elements of $\mathrm{Hol}(G)$, with g, $h \in G$, α, $\beta \in \mathrm{Aut}(G)$, the operation of $\mathrm{Hol}(G)$ is $(g,\alpha)(h,\beta) = (g+\alpha(h), \alpha\circ\beta)$. The identity element of this group is $(0,1)$, where 1 denotes the identity map on G, and the inverse of the element (g,α) is

$(g,\alpha)^{-1} = (-\alpha^{-1}(g), \alpha^{-1})$. As usual, we identify $\operatorname{Aut}(G)$ with the subgroup $\{(0,\alpha) \mid \alpha \in \operatorname{Aut}(G,+)\}$ of $\operatorname{Hol}(G,+)$,

There is an action of $\operatorname{Hol}(G,+)$ on $(G,+)$ by means of

$$(g,\alpha) * h = g + \alpha(h), \qquad g, h \in G, \quad \alpha \in \operatorname{Aut}(G,+).$$

We note that if $(G,+)$ is the additive group of a vector space, then $\operatorname{Hol}(G,+)$ can be identified with the affine group of the vector space G and this action corresponds to the natural action of the affine group on G. We observe that this action is faithful and so we can identify $\operatorname{Hol}(G,+)$ with a subgroup of the group Σ_G of all permutation of G. Given $g \in G$, the *left translation* of G given by g is defined by $\tau_g(h) = g + h$ for g, $h \in G$. The set T of all left translations defines a subgroup of Σ_G isomorphic to G. The action of τ_g coincides with the action of $(g,1) \in \operatorname{Hol}(G,+)$ on G. The next result is well known.

Proposition 2.1. *The normaliser in Σ_G of T coincides with the holomorph of $(G,+)$.*

When we speak about regular subgroups of the holomorph of a group, regularity is referred to the action defined above. Recall that a subgroup H of $\operatorname{Hol}(G,+)$ is *regular* when it acts transitively on G and the stabiliser H_g of an element $g \in G$ is trivial. This is equivalent to say that given h, $k \in G$, there exists a unique $(g,\alpha) \in H$ such that $(g,\alpha) * k = g + \alpha(k) = h$. Suppose that H is a regular subgroup of $\operatorname{Hol}(G,+)$. Then, given $h \in G$, there exists a unique $(g,\alpha) \in H$ with $g + \alpha(0) = h$. Consequently, $g = h$. If (h,α), $(h,\beta) \in H$, then $h + \alpha(0) = h + \beta(0)$ and the regularity of H shows that $\alpha = \beta$, that is, given $g \in G$ there exists a unique $\alpha = \lambda_g \in \operatorname{Aut}(G)$ with $(g,\alpha) \in H$. As $(0,1) * 0 = 0$, we have that $\lambda_0 = 1$ and so $H \cap \operatorname{Aut}(G,+) = \{(0,1)\}$.

Remark 2.2. *We can see other equivalent presentations of regular subgroups in the literature based on obtaining for an element $h \in G$ the unique element $(g,\alpha) \in H$ such that $(g,\alpha) * h = g + \alpha(h) = 0$, that is, given $h \in G$, there exists a unique $(g,\alpha) \in H$ with $g + \alpha(h) = 0$ (cf* [10, Lemma 4.1]*). We believe that our approach allows us to obtain in a more natural way the expression $H = \{(g,\lambda_g) \mid g \in G\}$ for a regular subgroup H of* $\operatorname{Hol}(G,+)$.

If $(G,+)$ is a finite group, an application of the orbit-stabiliser theorem to the action of a subgroup of $\operatorname{Hol}(G,+)$ on G yields the following result.

Proposition 2.3. *Let $(G,+)$ be a finite group and let H be a subgroup of* $\operatorname{Hol}(G,+)$. *Let us denote by π_G the "projection" of* $\operatorname{Hol}(G,+)$ *on G. Then $|H| = |\pi_G(H)||H \cap \operatorname{Aut}(G,+)|$.*

If we want to compute the regular subgroups of the holomorph of a finite group, Proposition 2.3 is useful to discard subgroups that cannot contain regular subgroups because the "projection" to G is not surjective. We can use the following result.

Proposition 2.4. *Let $(G, +)$ be a finite group and let H be a subgroup of $\operatorname{Hol}(G, +)$. The restriction of the "projection" π_G of $\operatorname{Hol}(G, +)$ to H is surjective if, and only if, $|H| = |G||H \cap \operatorname{Aut}(G, +)|$.*

The following characterisation of regular subgroups of $\operatorname{Hol}(G, +)$ for a finite group $(G, +)$ is also a consequence of Proposition 2.3.

Proposition 2.5 (cf. [1])**.** *Let $(G, +)$ be a finite group. Every two of the following three statements about a subgroup H of $\operatorname{Hol}(G, +)$ imply the other one.*

(1) $|H| = |G|$.
(2) $H \cap \operatorname{Aut}(G, +) = \{(0, 1)\}$.
(3) *The restriction to H of the "projection" π_G of $\operatorname{Hol}(G, +)$ on G is surjective.*

Moreover, a subgroup H of $\operatorname{Hol}(G, +)$ satisfying two of the three previous properties (and so the other one) is regular.

The following result can be obtained easily by computing $(g, \lambda_g)(k, \lambda_k)$.

Proposition 2.6. *Let H be a regular subgroup of $\operatorname{Hol}(G)$ and suppose that $H = \{(g, \lambda_g) \mid g \in G\}$. Then, given g, $k \in G$, we have that $\lambda_{g+\lambda_g(k)} = \lambda_g \circ \lambda_k$ and $\lambda_g^{-1} = \lambda_{\lambda_g^{-1}(-g)}$*

Bardakov, Neshchadim, and Yadav mention the following result in [6]. It is useful to obtain all skew left braces with a given additive group.

Proposition 2.7. *Let H be a regular subgroup of $\operatorname{Hol}(G, +)$ with $(G, +)$ a group and let $(g, \alpha) \in \operatorname{Hol}(G, +)$. Then $(g, \alpha)H(g, \alpha)^{-1}$ is again a regular subgroup of $\operatorname{Hol}(G, +)$. Furthermore, there exists $\beta \in \operatorname{Aut}(G, +)$ such that $(g, \alpha)H(g, \alpha)^{-1} = (0, \beta)H(0, \beta)^{-1}$.*

3. Regular subgroups and skew left braces

For the sake of completeness, we present here some well-known results useful to construct all skew left braces with a given additive group $(G, +)$. Some of them can be found in [10, Section 4] and in [3].

Proposition 3.1. *Let $(B, +, \cdot)$ be a skew left brace. Given $a \in B$, let $\lambda_a \colon B \longrightarrow B$ be the lambda map given by $\lambda_a(b) = -a + ab$. Then $H = \{(a, \lambda_a) \mid a \in B\}$ is a regular subgroup of $\operatorname{Hol}(B, +)$.*

Proposition 3.2. [10, Theorem 4.2] *Given a regular subgroup H of $\operatorname{Hol}(G, +)$, with $(G, +)$ a group, then H admits a structure of skew left brace whose additive group is isomorphic to $(G, +)$.*

The following result combines [6, Lemma 2.1 and Theorem 2.2].

Proposition 3.3. *Two regular subgroups H_1 and H_2 of $\operatorname{Hol}(G, +)$ induce isomorphic skew left braces if, and only if, they are conjugate by an element of $\operatorname{Aut}(G, +)$.*

By Proposition 2.7 we can replace the the condition of Proposition 3.3 by conjugation in $\operatorname{Hol}(G, +)$.

http://dx.doi.org/10.22108/ijgt.2023.137825.1846

4. Algorithms to compute isomorphism classes of skew left braces with a given additive group

Our starting point is the algorithm of Guarnieri and Vendramin [10, Algorithm 5.1] to construct all skew left braces with additive group $(G, +)$.

Algorithm 1. *To compute all isomorphism classes of skew left braces with a given additive group $(G,+)$, we proceed as follows:*

(1) *Compute the holomorph* $\mathrm{Hol}(G,+)$ *of* $(G,+)$*.*
(2) *Compute a list of representatives of the* $\mathrm{Aut}(G,+)$*-conjugacy classes of regular subgroups of* $\mathrm{Hol}(G,+)$*, all of them of order* $|G|$*.*
(3) *For each representative H of a* $\mathrm{Aut}(G,+)$*-conjugacy class of regular subgroups of* $\mathrm{Hol}(G,+)$*, we consider the map $p\colon G \longrightarrow H$ given by $a \longmapsto (f, f(a)^{-1})$, where $(f, f(a)^{-1}) \in H$. By defining a multiplication via $a \circ b = p^{-1}(p(a)p(b))$ for all a, $b \in G$, we obtain that $(G, +, \circ)$ becomes a skew left brace.*

Bardakov, Neshchadim, and Yadav have modified this algorithm in [6] by replacing $\mathrm{Aut}(G,+)$-conjugacy classes by simply conjugacy classes with the help of Proposition 2.7. If $(G,+)$ is a p-group for a prime p, then every regular p-subgroup of $\mathrm{Hol}(G,+)$ is conjugate in $\mathrm{Hol}(G,+)$ to a regular subgroup contained in a fixed Sylow p-subgroup of $\mathrm{Hol}(G,+)$. There are interesting algorithms to compute representatives of the conjugacy classes of subgroups of soluble groups, like the algorithm of Hulpke to compute the lattice of subgroups of a finite soluble group S. This algorithm is implemented, for instance, in the function `SubgroupsSolvableGroup` of GAP [9]. We summarise it here.

Algorithm 2. *To compute a list of representatives of the conjugacy classes of subgroups of a soluble group S, we proceed as follows:*

(1) *We compute a normal series $S \trianglerighteq N_1 \trianglerighteq \ldots \trianglerighteq N_r = 1$ with elementary abelian factors.*
(2) *By induction, we construct the subgroups of S/N_{i+1} from the subgroups of S/N_i. To simplify the description, we assume without loss of generality that $N_{i+1} = 1$, $N = N_i$, and that we know the subgroups of S/N. We have the following possibilities for a subgroup U of S:*
    (a) *U contains N, that is, U is the full preimage of a subgroup of S/N, already known, under the natural epimorphism;*
    (b) *U is contained in N and so U is a subspace of the vector space N, or*
    (c) *$B := U \cap N$ is a proper subgroup of N and $A = NU$ is a subgroup of G that contains properly N.*

    The subgroups of the first type can be computed by induction, and the subgroups of the second type are simply the subspaces of the vector space N. Thus it suffices to analyse the third case. Here $B \trianglelefteq U$ and $B \trianglelefteq N$, and so $B \trianglelefteq NU = A$ and $A \leqslant \mathrm{N}_S(A) \cap \mathrm{N}_S(B)$. We conclude that U/B is a complement of N/B in A/B.

http://dx.doi.org/10.22108/ijgt.2023.137825.1846

In the case that $G \cong C_4 \times C_4 \times C_4$, we have that $S = \mathrm{Hol}(G)$ is not soluble and we cannot apply directly this argument to G, but we can apply it to a fixed Sylow 2-subgroup S_2 of G. For $G \cong C_2 \times C_2 \times C_4 \times C_4$, we have that $S = \mathrm{Hol}(G)$ is soluble and so we can apply Hulpke's algorithm to S. Anyway, one of the problems of this algorithm is the big number of intermediate subgroups that can appear when running the algorithm, that can affect both the computing time and the memory required for the calculation. This was probably the reason behind the fact that Bardakov, Neshchadim, and Yadav were not able to compute the left braces with these additive groups in [6]. Since all regular subgroups of $\mathrm{Hol}(G,+)$ have the order $|G|$, it is useful to use some restrictions that can be imposed to the current implementation of `SubgroupsSolvableGroup`, like `ExactSizeConsiderFunction`, that avoids the computation of intermediate subgroups that do not lead to subgroups of the same order as G. We can modify the implementation of the algorithm to include also the restrictions stated in Proposition 2.3 that would eventually avoid the calculation of some intermediate subgroups. In the case of $G \cong C_4 \times C_4 \times C_4$, one of the challenges we have faced was the fact that two regular subgroups of $S = \mathrm{Hol}(G,+)$ contained in its Sylow 2-subgroup S_2 can be conjugate in S, but not in S_2. This makes necessary to search for different S_2-conjugacy classes that correspond to the same S-conjugacy class. The big number of S_2-conjugacy classes (31,367,678) and the corresponding checks for conjugation needed for this task pose another computational challenge.

We show here some of the solutions that we have given to these problems.

(1) The algorithm of Hulpke saves in the memory the list of representatives of all conjugacy classes of subgroups of G/N_i (we will refer to them as subgroups of the layer i) and the computed conjugacy classes of G/N_{i+1} on layer $i+1$. However, only one group of layer i is used at a time. Furthermore, the subgroups that appear as "descendants" of two different subgroups of a given layer in the next layers cannot coincide. We have used two proposals to avoid occupying a big amount of memory when there are many subgroups at a given layer. The first one, used in [3] for $G \cong C_4 \times C_4 \times C_4$, is to save each of the groups obtained at each layer to the hard disk and to recover it at the next layer. Of course, this has the drawback of the time needed to write and read the files, but in the event of a power interruption we could resume the computation by using these files. The second one, used in [5] for $G \cong C_2 \times C_2 \times C_4 \times C_4$, is to distribute the list of subgroups on one layer into several sublists and to apply our modification of Hulpke's algorithm to these sublists from the corresponding layer. The list of regular subgroups will appear as the union of the "descendants" of the members of the corresponding sublists. This is a problem that can clearly be parallelised.

(2) In the case of $G \cong C_4 \times C_4 \times C_4$, we have started by classify the regular subgroups by their isomorphism class, the isomorphism class of the kernel of the λ action of the brace on the additive group (that is, the set of all elements of the regular subgroup that stabilise all elements of $C_4 \times C_4 \times C_4$, that coincides in turn with the centraliser of the normal subgroup $C_4 \times C_4 \times C_4$

in the holomorph as a semidirect product with respect to this action), the isomorphism class of the quotient by this normal subgroup, and the length of the conjugacy class in $\mathrm{Hol}(G,+)$. This classifies all these conjugacy classes into 2,353 equivalence classes, as in [3]. It is clear that this comparison problem can also be parallelised.

For the case of $G \cong C_4 \times C_4 \times C_4$, we have started by implementing it on a computer with an Intel processor i7-11700 that allows the execution of 16 parallel tasks and 32 Gb of RAM running GNU/Linux during a couple of months. The partial results made us estimate that we would need more than two years to perform the task of identifying the $\mathrm{Hol}(G,+)$-conjugacy classes. We asked the Computer Service of the Universitat de València to allow us to do the computations on the scientific supercomputer *Lluís Vives* [8]. We are very grateful to the Computer Service for giving us immediate access to the machine, installing GAP, and solving our doubts. We could run these computations in parallel for each of the equivalence classes we have mentioned. Some of these equivalence classes were very large, for instance, four of them had more than one million subgroups. For these ones, we have started by dividing the list into many small sublists and then by removing $\mathrm{Hol}(G)$-conjugate subgroups in these sublists. The results of the comparisons have been saved in some lists that were later compared in pairs until only one representative of each $\mathrm{Hol}(G)$-conjugacy was obtained in the final step. The total number of conjugacy classes of regular subgroups of $\mathrm{Hol}(C_4 \times C_4 \times C_4)$ turns out to be 1,515,429, and this will be the number of left braces with additive group isomorphic to $C_4 \times C_4 \times C_4$.

For the case of $G \cong C_2 \times C_2 \times C_4 \times C_4$, we only must take care of applying our modification of Hulpke's algorithm. After noting that on layer 9 (of 11) we had obtained more than 50 million subgroups, we decided to split the lists of subgroups at each layer into several sublists and to compute the "descendants" of the groups on each sublist. In the case we obtain a large number of subgroups at another layer, we can split again the list. We have used also the *Lluís Vives* supercomputer to do this. The final step of the computations used 1505 jobs, from which we obtained 10,326,821 conjugacy classes of regular subgroups. In our case, there are 1,593,095,679 subgroups at layer 10. This big number of subgroups can pose a problem on current systems with 32 or 64 Gb, that could exhaust the physical memory even if an efficient representation of the regular subgroups is found.

The parallelisation of the problem has been reduced to running several copies of the algorithm with different parameters, we have not used any particular implementation of GAP with parallelism. As the rules to use *Lluís Vives* impose a limit of 96 clock hours for each computation, we have considered this implementation sufficiently suitable for our purposes. In case of long computations that could need more time to give a result, we have saved the partial results immediately before the 96 hours so that the algorithm could be resumed from that state.

TABLE 1. Left braces of order 64 by additive group

Group id	Structure	Number
$(64, 1)$	C_{64}	10
$(64, 2)$	$C_8 \times C_8$	11,354
$(64, 26)$	$C_4 \times C_{16}$	2,724
$(64, 50)$	$C_2 \times C_{32}$	142
$(64, 55)$	$C_4 \times C_4 \times C_4$	1,515,429
$(64, 83)$	$C_2 \times C_4 \times C_8$	743,410
$(64, 183)$	$C_2 \times C_2 \times C_{16}$	3,124
$(64, 192)$	$C_2 \times C_2 \times C_4 \times C_4$	10,326,821
$(64, 246)$	$C_2 \times C_2 \times C_2 \times C_8$	253,350
$(64, 260)$	$C_2 \times C_2 \times C_2 \times C_2 \times C_4$	2,189,661
$(64, 267)$	$C_2 \times C_2 \times C_2 \times C_2 \times C_2 \times C_2$	58,558
Total		15,095,601

5. Summary of results

The results of the computation and all left braces of order 64 obtained in [3, 5, 6] can be found on https://github.com/RamonEstebanRomero/braces64 [4] for use with GAP. The storage of these left braces is based on the ideas of [2] on the interpretation of a left brace as a triply factorised group. We have stored a left brace as the semidirect product of its multiplicative and its additive group, a triply factorised group that is soluble, by means of a number that encodes the pc-presentation of a finite soluble group like in the function `CodePcGroup` of GAP [9]. We have had a similar idea to encode the diagonal-type subgroup of the triply factorised group. The repository also includes some functions that allow us to work with these left braces with the help of the YangBaxter package [13] by Vendramin and Konovalov.

The total number of left braces of order 64 is 15,095,601. The numbers of left braces of order 64 by additive group, that combine the results in [3, 5, 6], can be found on Table 1.

Acknowledgements

We thank the Computer Service of the Universitat de València for allowing us to use the scientific supercomputer *Lluís Vives* and for their kind support in setting up the system for the computation and solving our questions.

These results are part of the R+D+i project supported by the Grant PGC2018-095140-B-I00, funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".

We thank the anonymous referees for their comments, that have helped us to improve the presentation of the paper.

References

[1] E. Acri and M. Bonatto, Skew braces of size p^2q I: Abelian type, *Algebra Colloq.*, **29** (2020) 297–320.

[2] A. Ballester-Bolinches and R. Esteban-Romero, Triply factorised groups and the structure of skew left braces, *Commun. Math. Stat.*, **10** (2022) 353–370.

[3] A. Ballester-Bolinches, R. Esteban-Romero and V. Pérez-Calabuig, Enumeration of left braces with additive group $C_4 \times C_4 \times C_4$, *Math. Comp*, (in press), (2023) https://doi.org/10.1090/mcom/3871}.
"bibitemBallesterEstebanPerezC22-braces64-GitHub ———, Left braces of order 64, 2022, https://doi.org/10.5281/zenodo.7406729.

[4] ———, Enumeration of left braces with additive group $C_2 \times C_2 \times C_4 \times C_4$, *Ric. Mat.*, (2023), https://doi.org/10.1007/s11587-023-00772-2.

[5] V. G. Bardakov, M. V. Neshchadim and M. K. Yadav, Computing skew left braces of small orders, *Internat. J. Algebra Comput.*, **30** (2020) 839–851.

[6] W. Bosma, J. J. Cannon, C. Fieker, and A. Steel, *Handbook of magma functions*, Version 2.19, 2016.

[7] Computer Service, Universitat de València, LluisVives v2 user's guide, Universitat de València, 2020, https://www.uv.es/calcul/Manuales/LluisVives2_user_manual.pdf, visited 8th April, (2022).

[8] *The GAP Group*, GAP Groups, Algorithms, and Programming, Version 4.11.1, 2021.

[9] L. Guarnieri and L. Vendramin, Skew-braces and the Yang-Baxter equation, *Math. Comp.*, **86** (2017) 2519–2534.

[10] W. Rump, Braces, radical rings, and the quantum Yang-Baxter equation, *J. Algebra*, 307 (2007) 153–170.

[11] L. Vendramin, Problems on skew left braces, *Adv. Group Theory Appl.*, **7** (2019) 15–37.

[12] L. Vendramin and O. Konovalov, YangBaxter: combinatorial solutions for the Yang-Baxter equation, 2022, version 0.10.2, https://www.gap-system.org/Packages/yangbaxter.html.

Adolfo Ballester-Bolinches

Departament de Matemàtiques, Universitat de València, Dr. Moliner, 50, 46100, Burjassot, València, Spain

Email: Adolfo.Ballester@uv.es

Ramón Esteban-Romero

Departament de Matemàtiques, Universitat de València, Dr. Moliner, 50, 46100, Burjassot, València, Spain

Email: Ramon.Esteban@uv.es

Vicent Pérez-Calabuig

Departament de Matemàtiques, Universitat de València, Dr. Moliner, 50, 46100, Burjassot, València, Spain

Email: Vicent.Perez-Calabuig@uv.es

http://dx.doi.org/10.22108/ijgt.2023.137825.1846