



A Look at Algorithm $BEPtoPNS_T^*$

Juan C. García-Ojeda**

Received: 30/03/2020 • Accepted: 09/09/2020

<https://doi.org/10.22395/rium.v20n39a7>

Abstract

This work analyzes the computational complexity of algorithm $BEPtoPNS_T$, which transforms a building-evacuation problem (BEP) into a time-expanded, process-network synthesis (PNS_T) problem. The solution of the latter is achieved by resorting to the P -graph method which exploits the combinatorial nature of a BEP . Unlike other approaches, the P -graph method provides not only the optimal solution (best evacuation route as a function of egress time), but also the best n sub-optimal solutions. For the complexity analysis, a generic processor, and a Random-access machine (RAM) model were deployed as well as a mathematical model to calculate the number and cost of the operations performed. It was observed that algorithm $BEPtoPNS_T$ exhibits an asymptotic complexity of order $O(T|A|(|N|-k))$. When solving a BEP , however, the total complexity grows exponentially with order $O(T|A|(|N|-k) + O(2^h))$ in the worst case; where h represents the total number of operating units specified in the corresponding PNS_T problem. Nevertheless, the computational complexity can be reduced significantly when the P -graph method is deployed.

Keywords: building-evacuation routes; process-network synthesis; rigorous structures; combinatorial optimization; P -graph; pseudocode; RAM model; algorithm complexity; asymptotic notation; Big- O .

* This work is based on the second chapter of the doctoral dissertation entitled “On Modeling Building-Evacuation-Route Planning and Organization-based Multiagent Systems by Resorting to the P-graph Framework” [13].

** Programa de Ingeniería de Sistemas, Facultad de Ingeniería, Universidad de Cartagena. Avenida del Consulado No. 48 – 152, Bloque G, Oficina G4-136. Email: jgarcia@unicartagena.edu.co. Orcid: <https://orcid.org/0000-0002-3148-0436>

Una mirada al algoritmo BEPtoPNS_T

Resumen

El presente trabajo estudia y analiza la complejidad computacional, en el peor de los casos, del algoritmo $BEPtoPNS_T$. El objetivo de $BEPtoPNS_T$ es transformar problemas de rutas de evacuación en edificios (*Building-Evacuation Problems, BEP*) en problemas de síntesis de redes procesos de tiempo extendido (*Time-Extended, Process-Network Synthesis, PNS_T*), los cuales se solucionan por medio del método *P-graph*. La relevancia de analizar el algoritmo $BEPtoPNS_T$ se sustenta en el hecho que el método *P-graph* explota la naturaleza combinatoria de un *BEP* luego de ser transformado en un PNS_T . El método *P-graph* no sólo provee la solución óptima (mejor ruta de evacuación en función del tiempo de egreso), sino que también provee las mejores n soluciones subóptimas; característica que a la fecha no ofrecen otros métodos de optimización. Para el análisis del algoritmo $BEPtoPNS_T$, se consideró un procesador genérico, el modelo *Random-access machine, RAM*, así como un modelo matemático para calcular el número de operaciones ejecutadas y sus costos, resultando en una complejidad asintótica de orden $O(T | A | (|N| - k))$. Sin embargo, la complejidad total del proceso, incluyendo el método *P-graph*, crece de manera exponencial, es decir, $O(T | A | (|N| - k)) + O(2^N)$, en el peor de los casos.

Palabras clave: rutas de evacuación en edificios; síntesis de redes de procesos; estructuras rigurosas; optimización combinatoria; P-graph; pseudocódigo; modelo RAM; complejidad de algoritmos; notación asintótica; O grande.

INTRODUCTION

Colombian standard NSR10 for earthquake-resistant buildings (*Reglamento Colombiano de Construcciones Sismo Resistentes* in its Spanish original) in its sections J and K describes the minimum safety requirements in the event of fire or any other event that threatens the safety of people inside buildings [1][2]. These requirements include (i) reducing fire risks or any other threat as much as possible; (ii) preventing fire hazards from reaching nearby buildings; (iii) facilitating people-evacuation tasks; (iv) facilitating mitigation of hazards; and (v) minimizing the risk of building collapse when evacuating people and extinguishing the threat [1]. Similarly, the Colombian government has adopted a National Policy for the Management of Disaster Risks (SNGRD) [3]; moreover, it has set out the guidelines for the construction of disaster risk management plans in public and private organizations (see *Decree 2157 of 2017*) [4]. These guidelines lay out the actions for effecting evacuation plans so that people inside a building (e.g., house, office, school, mall, etc.) can exit from it orderly, rapidly, and safely so that their lives are protected. Such actions concern (i) the identification of the building and its surroundings; (ii) protective measures for evacuations; (iii) evacuation paths; (iv) evacuation signage and warnings; and (v) identification of permanent and temporary safety points [5][6].

Although all these elements are important in the elaboration of an evacuation plan, the ultimate purpose of such a plan is to guarantee the rapid and safe movement of people away from any threat (e.g., fire) or accident (e.g., explosion) in the shortest possible time [7]. However, an evacuation plan may involve the evaluation of an exceedingly large number of evacuation routes in particular scenarios (e.g., buildings, city, etc.), which can be quite complex in view of the combinatorial nature of the problem [8-12]. Currently, research on evacuation routes comprises the following six categories [13]: level of service, mathematical models, heuristic methods, stochastic models, simulation tools, and multiagent systems. However, few works analyze the mathematical complexity of these techniques [14], a fundamental aspect when selecting one option over another. Specifically, in relation to the time and space used by the algorithms to solve the problem according to the size or length of the input [15]. Moreover, when these technologies, at most, generate a single solution to the problem of evacuation routes [16-18]. For this reason, this work analyzes the complexity, in terms of execution time, of algorithm $BEPtoPNS_T$ [19-24] with the aim to advance in the development of computational strategies that allow efficient planning of evacuation routes. The novel algorithm $BEPtoPNS_T$ allows not only to generate the optimal evacuation plan, but also sub-optimal plans [19-24], which uses a graph-theoretical approach based on *P-graphs* (process graphs) [25-31]. In what follows, the methods used are described, algorithm $BEPtoPNS_T$ is analyzed asymptotically, and finally, conclusions are drawn.

1. METHODS

1.1 Problem Definition

A building-evacuation problem (*BEP*) can be modeled as a directed graph [8]; the goal is to minimize the evacuation time. Let $G = (N, A)$ be a directed graph with N the set of nodes and A the set of arcs; each node $n \in N$ represents potential locations of evacuees and other areas on the building (e.g., rooms or stairs). Moreover, the possible movements between locations through k (e.g., corridors or doors) are represented by arcs $(i, j, k) \in A$, such that $i, j \in N$ (see figure 1); furthermore, G must satisfy the restrictions of the building [8]. Specifically, each location n has a limited capacity expressed by a non-negative integer, cap_n , representing the number of people that can be accommodated in n .

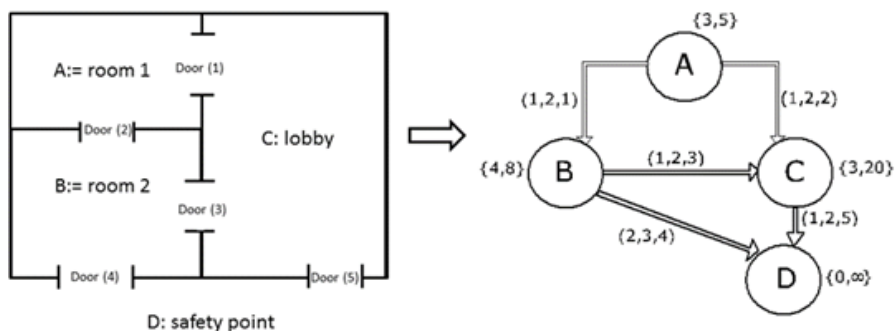


Figure 1. Conventional adopted graph-based notation for representing building-floor maps REF _Ref420414599 \r \h * MERGEFORMAT [8]: {initial contents, node capacity}; (travel time, arc capacity, arc id)

Source: taken from REF _Ref50116325 \r \h * MERGEFORMAT [20].

For each location, a non-negative integer is assigned to initial occupancy, ic_n , signifying the number of people at any location in the event of an emergency. In addition, a positive integer, $cap_{(i,j,k)}$, is assigned to the maximum flow rate of corridor (i,j,k) , which represents the maximum number of people that can pass through a corridor simultaneously. Finally, each corridor (i,j,k) is constrained by travel time $\lambda_{(i,j,k)}$, a non-negative integer that measures the time a person takes to traverse the total length of a passageway [8-12].

1.2 Process Graphs (P-graphs)

An innovative approach called *P-graph* has been proposed to facilitate the solution of process-network synthesis (*PNS*) problems [27]. A *P-graph* is a mathematically rigorous bipartite graph that allows the graphical representation of a *PNS* problem [32][33]. The *P-graphs* together with a solid axiomatic basis facilitate the construction of structures (flowsheets) of a given process by algorithmically combining operating units that are

part of the process. A *P-graph* is formally defined in equation (1) in terms of a set of operating units, O , as a subset of the cartesian product of all subsets of materials, M , as

$$O \subseteq \wp(M) \times \wp(M) \tag{1}$$

where $\wp(M)$ denotes the power set of M and \times , the cartesian product. Moreover, in equation (2), a *P-graph* is defined as a pair (M, O) whose vertices are elements of set V given by

$$V = M \times O \tag{2}$$

Vertices *M-type* and *O-type* belong to the M and O sets, respectively. The arcs of the graph are the elements described by the following expressions in equations (3), (4), and (5).

$$A = A1 \times A2 \tag{3}$$

where

$$A1 = \{(X,Y) \mid Y = (\alpha, \beta) \in O \wedge X \in \alpha\} \tag{4}$$

and

$$A2 = \{(Y,X) \mid Y = (\alpha, \beta) \in O \wedge X \in \beta\} \tag{5}$$

Thus, X indicates an *M-type vertex*; Y , an *O-type vertex*; α , a set of *M-type vertices*, of which the arcs are directed to the *O-type vertices*; and β , a set of *M-type vertices* whose arcs are directed to the *O-type vertices* [26]. For illustration, let M be a set of materials, $M = \{A,B,C,D,E,F\}$, and O a set of operating units given by $O = \{(\{B,A\}, \{A\}), (\{D,E\}, \{B,C\}), (\{F\}, \{A,C\}), (\{F\}, \{A,C\})\}$. It can be verified that M and O satisfy the restrictions (1-5), such that (M,O) is a *P-graph* as shown in figure 2 [27].

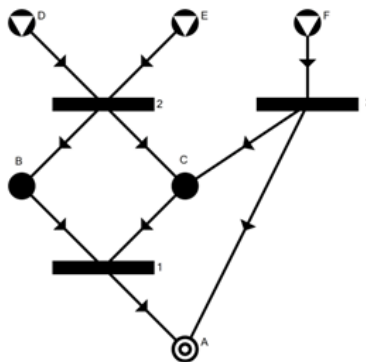


Figure 2. *P-graph* (M,O) where A,B,C,D,E, and F are materials, and 1,2, and 3 are the operating units: represents raw materials or input elements of the whole process; symbolizes intermediate-materials or elements, emerging between the operating units; and represents products or outputs of the entire process

Source: adapted from REF_Ref36019017 \r \h * MERGEFORMAT [27].

1.3 Analysis of Algorithms

Algorithm analysis plays a critical role in computer science as it guides the assessment of efficient procedures for addressing any given computational problem [34] [35]. This assessment is based on the execution time of the instructions, or steps, that an algorithm must follow without considering the impact of other factors [35].

Equation (6) embodies the mathematical model used to analyze the complexity of algorithm $BEPtoPNS_T$. At the outset, the basic instructions (q) of the algorithm are identified; then, their costs are determined ($cost_q$), and finally, the number of times that those operations are executed ($times_q$) is analyzed [36].

$$\sum_q cost_q \times times_q \quad (6)$$

Most algorithms work well with small inputs; however, we are interested in analyzing the behavior of the algorithm when the input size is large enough. To this end, we define the execution time of an algorithm as in equation (7).

$$f(n): N \rightarrow R^{\geq 0} \quad (7)$$

where $n \in N$ represents the size of the input, $f(n)$, the number of steps executed by the algorithm given n , and $R^{\geq 0}$, the set of non-negative real numbers. Therefore, attention is given to the behavior of $f(n)$ when n increases. The above expression allows to relate the asymptotic behavior of an algorithm through $f(n)$. Based on $f(n)$, functions $O(f(n))$, $\Omega(f(n))$, and $\Theta(f(n))$ are defined [15],[35-38], thus signifying, respectively, a function that provides an upper limit (worst case, maximum number of instructions performed by an algorithm), one that provides a lower limit (best case, minimum number of instructions performed by an algorithm), and a function that is contained between the upper and lower limits, in the order of growth of the function. The ideal is to analyze the asymptotic behavior in the worst case of any given algorithm [15],[35-38]; therefore, it is important to define the term, O . The notation O , $O(f(n))$ defines the set of all functions $g(n): N \rightarrow R^{\geq 0}$, such that for some natural number n_0 and some positive real number c , $g(n) \leq cf(n)$, as long as $n \geq n_0$ [35]. Informally, we say that the expression $g(n)$ is less than or equal to some constant multiple of $f(n)$.

2. ALGORITHM $BEPtoPNS_T$

Algorithm $BEPtoPNS_T$ transforms a BEP into a PNS_T problem, thus representing the temporal dimension of a BEP (i.e., evacuation time), which is solved with the aid of the P -graph method [25-31]. The algorithm consists of two parts: initialization and time expansion. During initialization, the materials that describe a PNS problem are identified. Such materials are determined from the nodes specified in the graph that

describes the floor map of a building. These materials can be (i) raw materials, (ii) intermediate materials, and (iii) products. The raw materials represent locations in the building where people are located just before the emergency alarm is activated; intermediate materials represent empty locations; and the products represent safety points, or meeting areas, outside of it. On the other hand, the time expansion identifies the operational units of a PNS problem, which describe the way in which people move during an evacuation plan. Such movements can be of two types: (i) from one location to another; or (ii) a momentary pause at the current location. A more detailed discussion on algorithm BEPToPNS_T is presented elsewhere [20]. Table 1 lists the elements involved in the specification of algorithm BEPToPNS_T, as well as the analysis of its complexity.

Table 1. Mathematical notation for specifying algorithm BEPToPNS_T.

Symbol	Description
R	set of raw materials, that is, the locations with people at the beginning of the evacuation in the original problem
r	element of R , i.e., $r \in R$
U_r	upper limit of r , that is, the number of people at location r at the start of the evacuation
L_r	lower limit of r , that is, 0
$ $	cardinality of a set, that is, the number of elements contained in it
P	set of products, that is, the exits or safe points outside the building once the evacuation of the building has begun
$\sum_{i=1}^{ R } U_i$	the total number of people inside the building at the start of the evacuation
T	maximum time expected to evacuate the building
t	unit of time
$M_{i,t+1}$	number of individuals accommodated in i at time $t + 1$
$O_{i,j,t,t+1}$	number of individuals who prefer to be in a specific location, i , at least one more unit of time, i.e., $t + 1$.
$L_{O_{i,j,t,t+1}}$	lower limit of $O_{i,j,t,t+1}$, that is, the number of people that can be at location i at least one more time unit, i.e., $t + 1$
$U_{O_{i,j,t,t+1}}$	upper limit of $O_{i,j,t,t+1}$, that is, the number of people that can be at location i at least one more unit of time, i.e., $t + 1$, that is cap_i
$O_{i,j,t,t+\lambda,j,k}$	number of individuals walking from location i at time t to location j through a corridor k at time $(i,j,k) + t$
$L_{O_{i,j,t,t+\lambda,j,k}}$	lower limit of $O_{i,j,t,t+\lambda,j,k}$, that is, the number of individuals that walk from location i at time t to location j through a corridor k at time $(i,j,k) + t$
$U_{O_{i,j,t,t+\lambda,j,k}}$	upper limit of $O_{i,j,t,t+\lambda,j,k}$, that is, the number of individuals who walk from location i at time t to location j through a corridor k at time $(i,j,k) + t$
$O_{evacuation,t}$	people who reach a safety point, k , at time t
$L_{O_{evacuation,t}}$	lower limit of people who will reach a safety point, k , at time t
$U_{O_{evacuation,t}}$	upper limit of people who reach a safety point, k , at time t
$cp_{O_{evacuation,t}}$	proportional cost associated with the number of people who reach a safety point, k , at time t

Source: own elaboration.

2.1 Complexity Analysis

As mentioned earlier, algorithm $BEPtoPNS_T$ comprises two parts, initialization (statements $st1$, $st2$ and loop $lp1$, lines 4-11) and time expansion (statement $st3$ and loops $lp2$ and $lp3$, lines 12-53). Initially, for each node $n \in N$ in G where $ic_n > 0$, it is transformed into a raw material r and is added to set R (lines 4 and 8). Initially, in loop $lp1$, for each raw material, r , the lower limit, L_r , and the upper limit, U_r , are set, such that the total amount of materials available for the problem PNS_T is defined. The complexity of the statement $st1$ is given by the number of nodes $n \in N$ that meet the condition $ic_n > 0$ and the time required by loop $lp1$, which in the worst case is bounded by the expression $|N| - k + 1$ nodes. The above under the assumption that $|N| - k$ represents locations that at the beginning of the emergency shelter people and k safety points outside the building ($k > 1$), where $||$ represents set cardinality. Finally, the product, $SafetyPoint$, is specified and added to the set P (statement $st2$, line 11). For the product, $P_{SafetyPoint}$, algorithm $BEPtoPNS_T$ sets the lower limit $L_{P_{SafetyPoint}}$ and upper limit $U_{P_{SafetyPoint}}$; such that the quantity of products to be manufactured is equal to the quantity of raw material demanded by the problem PNS_T . The number of people inside the building at the start of the emergency, $\sum_{i=1}^{|R|} U_i$, must be equal to the number of people evacuated (product, $P_{SafetyPoint}$). Because the statement $st2$ describes assignments, its execution time is constant [15], [35-38] (see figure 3).

By assuming a limit to the evacuation time, T , algorithm $BEPtoPNS_T$ generates materials $M_{i_{(t+1)}}$ for $0 \leq t < T$; consequently, loop $lp2$ is executed $T + 1$ times (line 13). The material $M_{i_{(t+1)}}$ represents the number of individuals present in i at time $t + 1$ (lines 18-19) ([8], p. 7). Subsequently, an operating unit o is created and added to set O for each i and t and operating units $O_{i_{i_{i_{(t+1)}}$ are generated, where $0 \leq t < T$ (lines 20-23). Similarly, the lower limit $L_{O_{i_{i_{i_{(t+1)}}$ and upper limit $U_{O_{i_{i_{i_{(t+1)}}$ are set for each operating unit $O_{i_{i_{i_{(t+1)}}$, such that algorithm $BEPtoPNS_T$ specifies the number of individuals who prefer to stay in the specific location i for at least one unit of time, $t + 1$ ([8], p. 7). These instructions are executed $T * (|N| - k)$ times. Subsequently, node j is transformed into material m and added to set M for each arc $(i, j) \in A$ in G , where the value of $cap_j \neq \infty$ or j is the only safety point (lines 24, 28-33). These operations are executed $T * (|N| - k) * (|A| - 1)$ times. Then, for each t , i , and j , operating unit o is created and added to O ; algorithm $BEPtoPNS_T$ generates operating units $O_{i_{j_{i_{(t+\lambda_{ijk})_k}}$, where $0 \leq t < T$. In addition, the lower limit $L_{O_{i_{j_{i_{(t+\lambda_{ijk})_k}}$ and upper limit $U_{O_{i_{j_{i_{(t+\lambda_{ijk})_k}}$ are set for each operating unit $O_{i_{j_{i_{(t+\lambda_{ijk})_k}}$ (lines 34-37), such that algorithm $BEPtoPNS_T$ specifies the number of evacuees moving from location i at time t to location j through corridor k at time $t + \lambda_{ijk}$ ([8], p. 7). Similarly, these instructions are executed $T * (|N| - k) * (|A| - 1)$ times.

Finally, loop lp3 (line 43) specifies the operating units representing the number of evacuees who reach a safety point, i.e., *SafetyPoint*. First, the algorithm iterates over $0 < t \leq T$ (line 43); this instruction is executed $T + 1$ times. As a result, the operating units, $O_{evactime_t}$, are generated for each safety point, k ; this instruction is executed $T * k$ times (line 48). Likewise, the lower limit $L_{O_{evactime_t}}$, upper limit $U_{O_{evactime_t}}$, and the proportional cost $cp_{O_{evactime_t}}$ of each operating unit $O_{evactime_t}$ are set, such that the number of evacuees who reach a safety point at time t is specified. Figure 4 shows the relationship of the elements adopted in the definition of a BEP with those adopted in the specification of a PNS_T problem. Based on equations (6) and (7), equation (8) expresses the execution time $g(G,T)$ of algorithm $BEPtoPNS_T$.

$$\begin{aligned}
 g(G,T) = & (T-1)(|N|-k)(|A|-1)(c_{25}) \\
 & + T(|N|-k)(|A|21)(c_{20} + c_{21} + c_{22} + c_{23} + c_{24} + c_{26} + c_{27} + c_{28} + c_{29}) \\
 & + T(|N|-k)A(c_{17} + c_{19}) + T(|N|-k)(c_{11} + c_{12} + c_{13} + c_{14} + c_{15} + c_{16} + c_{18}) \quad (8) \\
 & + T(|N|-(k+1))(c_{10}) + Tk(c_{32} + c_{33}) + T(c_8 + c_9 + c_{30} + c_{31} + c_{35}) + k(c_{32} + c_{33}) \\
 & -k(c_2 + c_3 + c_4 + c_5) + N(c_1 + c_2 + c_3 + c_4 + c_5 + c_7) + (c_1 + c_2 + c_6 + c_7 + c_8 + c_{30} + c_{32} + c_{34})
 \end{aligned}$$

where clearly execution time g depends on G and T . Assuming that costs c_i are constant [15],[35-38] and rearranging the terms of equation (8), equation (9) yields

$$g(G,T) = 12[T|A|(|N|-k)] - 2T(7k+|N|) - |A|(|N|-k) + 7N - 3k + 4T + 6 \quad (9)$$

By applying equation (9) to any size of G and T with constant $c \geq 12$, the complexity of $BEPtoPNS_T$ can be summarized in equation (10).

$$g(G,T) \leq O(T|A|(|N|-k)) \quad (10)$$

which is the expression of complexity for the worst case. It should be noted, as mentioned in [39], that the computational complexity of a PNS problem is of exponential order given by $O(2^h)$; where h represents the number of operating units. Thus, equation (11) shows that complexity grows exponentially with order $O(T|A|(|N|-k)) + O(2^h)$ in the worst case when solving a BEP .

$$g(G,T) \leq O(T|A|(|N|-k)) + O(2^{|h|}), \quad (11)$$

where h is the sum of $|O_{i_i_t(t+1)}|$, $|O_{i_j_t(t+\lambda_{ijk})-k}|$, and $|O_{evactime_t}|$.

3. DISCUSSION

Algorithm $BEPtoPNS_T$ aims at transforming a BEP into a PNS_T problem by resorting to the P -graph method. However, it must be considered that the complexity involved in solving PNS problems is of exponential order as mentioned in the preceding section. Formally, it has been shown that the complexity of a PNS problem is equivalent to the

set covering problem [40]. Because the set covering problem is one of Karp's original 21 NP-complete problems, it can be inferred that the *PNS* problem is NP-complete [41]. This sort of problem can be addressed by using the *P-graph* method [27] because the method can drastically reduce the search space of a given *PNS* problem [27], [29].

Despite the computational cost of solving a *BEP* as a PNS_T problem (see equation (11)), the latter can be solved by resorting to the *P-graph* method which exploits the combinatorial nature of a *BEP*. Unlike other approaches, such as Hoppe and Tardos or algorithms CCRP and SSEP [14], the *P-graph* method provides not only the optimal solution (best evacuation route as a function of egress time), but also the best n sub-optimal solutions [19]. In addition, the *P-graph* method can also model multiple starting locations and safety points, a feature not supported in SSEP [14]. The foregoing invites us to continue the work of reducing the computational complexity of solving PNS_T problems in relation to *BEPs*.

4. CONCLUSIONS

This work analyzed the complexity of algorithm $BEPtoPNS_T$, which was found to be of order $O(T|A|(|N|-k))$ in the worst case with $k > 1$ safety points outside a building. Moreover, the algorithm's computational complexity grows exponentially with order $O(T|A|(|N|-k)) + O(2^{|h|})$ when solving a *BEP*, in the worst case. These results raise issues that deserve further research. First, the complexity of the entire process for finding at least the optimal solution to a *BEP* must be evaluated in terms of the minimum number of steps involved. The *P-graph* method with its five (5) fundamental axioms is able of reducing the search space; however, its complexity is of order $O(2^{|h|})$ in the worst-case scenario. Therefore, it is required to analyze the average complexity [15], [35]. Second, it will be essential to design real-time monitoring systems for evacuation-route planning; such systems could require the adoption of modern computing techniques, e.g., Grid Computing, High-Performance Computing (HPC), and Internet of Things (IoT), to significantly accelerate its computation [42-44]. Finally, it will be necessary to evaluate the adoption of the strategy described in this work in other engineering fields. For instance, the design of Very-Large-Scale Integration (VLSI) circuits commonly faces problems that require the determination of optimal routes between the circuits that interconnect various electronic components, thereby implying the minimization of wiring length [45-48].

5. ACKNOWLEDGEMENTS

The author is grateful to the Faculty of Engineering of Universidad de Cartagena for its financial support as well as to Dr. Andrés Argoti for his editorial review of the manuscript.

REFERENCES

- [1] Ministerio de Ambiente, Vivienda, y Desarrollo Territorial, Reglamento Colombiano de Construcción Sismo Resistente – NSR10: Capítulo J – Requisito de Protección contra Incendios en Edificaciones. Colombia, 2010, pp. J1-J32.
- [2] Ministerio de Ambiente, Vivienda, y Desarrollo Territorial, Reglamento Colombiano de Construcción Sismo Resistente – NSR10: Capítulo K – Requisitos Complementarios. Colombia, 2010, pp. K1-K64.
- [3] Ley 1523 de 2012. Congreso de la República de Colombia, Bogotá, Colombia, April 2012.
- [4] Decreto 2157 de 2017. Departamento Administrativo de la Presidencia de la República. Bogotá, Diciembre de 2017.
- [5] Unidad Nacional para la Gestión de Riesgos y Desastres – Colombia. Guía para la Elaboración de Planes de Evacuación (2da. Ed.), Status Publicidad, Bogotá, 2016, pp. 1-17.
- [6] ICONTEC. Norma Técnica Colombiana – NTC 1461, Instituto Colombiano de Normas Técnicas y Certificación, Bogotá, 1987, pp. 1-18.
- [7] W.H. Stringfield, *Emergency Planning and Management (2nd Edition)*. Lanham, MD: Government Institutes, 2000, pp. 1-294.
- [8] H.W. Hamacher and S.A. Tjandra, “Mathematical modeling of evacuation problems: a state of the art,” in *Pedestrian and Evacuation Dynamics*, M. Schreckenberg and S.D. Sharma, eds., pp. 227–266, Berlin: Springer, 2002.
- [9] M. Skutella, “An introduction to network flows over time,” in *Research Trends in Combinatorial Optimization*, W. Cook *et al.*, eds., pp. 451–482, Berlin: Springer, 2009.
- [10] S. Kim *et al.*, “Contraflow transportation network reconfiguration for evacuation route planning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, n° 8, pp. 1115–1129, 2008. DOI: 10.1109/TKDE.2007.190722
- [11] E.D. Kuligowski, and R.D. Peacock, *A Review of Building Evacuation Models (1st ed.)*, National Institute of Standards and Technology, Technical Note 1471, 2005.
- [12] E.D. Kuligowski *et al.*, *A Review of Building Evacuation Models (2nd ed.)*, National Institute of Standards and Technology, Technical Note 1680, 2010.
- [13] J.C. García-Ojeda. *On Modeling Building-Evacuation-Route Planning and Organization-based Multiagent Systems by Resorting to the P-graph Framework*, PhD Dissertation, University of Pannonia, Veszprem, Hungary, 2016.
- [14] G. Mishra *et al.*, *Improved Algorithms for the Evacuation Route Planning Problem*. In: Lu Z., Kim D., Wu W., Li W., Du DZ. (eds) *Combinatorial Optimization and Applications*. Lecture Notes in Computer Science, vol. 9486. Springer, 2015.

- [15] T.H. Cormen *et al.*, *Introduction to Algorithms*, 3a ed., Cambridge: MA, MIT Press, 2009, pp. 1-1320.
- [16] L. Han *et al.*, “An Efficient Staged Evacuation Planning Algorithm Applied to Multi-Exit Buildings,” in *Int. J. Geo-Inf.*, vol. 9, n° 46, 2020.
- [17] X. Wang and H. Liu, “An evacuation algorithm for large buildings,” in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, Beijing, 2012, pp. 2497-2502, DOI: 10.1109/WCICA.2012.6358293
- [18] C.H. Oh *et al.*, “An Efficient Building Evacuation Algorithm in Congested Networks,” in *IEEE Access*, vol. 7, pp. 169480-169494, 2019, DOI: 10.1109/ACCESS.2019.2955477
- [19] J. C. Garcia-Ojeda *et al.*, “Modeling and Assessing Evacuation Route Plans by Resorting to the P-graph Framework,” *2020 Smart City Symposium Prague (SCSP)*, Prague, Czech Republic, 2020, pp. 1-6, DOI: 10.1109/SCSP49987.2020.9134050
- [20] J.C. Garcia-Ojeda *et al.*, “Building-Evacuation-Route Planning via Time-Expanded Process-Network Synthesis,” *Fire Safety Journal*, vol. 61, pp. 338–347, 2013. DOI: 10.1016/j.firesaf.2013.09.023
- [21] J.C. Garcia-Ojeda *et al.*, “Planning evacuation routes with the P-graph framework,” *Chemical Engineering Transactions*, vol. 29, pp. 1531-1536, 2012. DOI: 10.3303/CET1229256
- [22] J.C. García-Ojeda *et al.*, “Identifying Evacuation Routes via the P-graph Methodology,” presented at the 10th Colombian Computation Congress, Bogotá, 2015.
- [23] J.C. García-Ojeda *et al.*, “Multi-criteria Analysis of Building Evacuation Route Planning by Resorting to the P-graph Framework,” presented at 5th Veszprem Optimisation Conference: Advanced algorithms, Veszprem, 2012.
- [24] J.C. Garcia-Ojeda, “On Building Evacuation Route Planning by Resorting to P-graph,” *Revista Colombiana de Computación*, vol. 12, n° 1, pp. 111–125, 2011.
- [25] F. Friedler *et al.*, “Combinatorially Accelerated Branch-and-Bound Method for Solving the MIP Model of Process Network Synthesis,” in *Global Optimization, Computational Methods and Applications, State of the Art*, C.A. Floudas and P.M. Pardalos, eds., pp. 609-626, Dordrecht, Netherlands: Kluwer Academic Publishers, 1996.
- [26] F. Friedler *et al.*, “Decision-mapping for design and synthesis of chemical processes: applications to reactor-network synthesis,” In *Foundations of Computer-Aided Process Design, AIChE Symposium Series 91*, L. T. Biegler and M. F. Doherty, eds. pp. 246 – 250, NY: American Institute of Chemical Engineers, 1995.
- [27] F. Friedler *et al.*, “Graph-theoretic approach to process synthesis: axioms and theorems,” *Chemical Engineering Science*, vol. 47, n° 8, pp. 1972–1988, 1992. DOI: 10.1016/0009-2509(92)80315-4

- [28] F. Friedler *et al.*, “Combinatorial Algorithms for Process Synthesis,” *Computers & Chemical Engineering*, vol. 16, n° 5, pp. S313–320, 1992. DOI: 10.1016/S0098-1354(09)80037-9
- [29] F. Friedler *et al.*, “Process Network Synthesis: Problem Definition.” *Networks*, vol. 28, n° 2, pp. 119–124, 1998. DOI: 10.1002/(SICI)1097-0037(199803)31:2<119::AID-NET6>3.0.CO;2-K
- [30] J.J. Klemeš and P.S., Varbanov, “Spreading the message: p-graph enhancements: implementations and applications”, *Chemical Engineering Transactions*, vol. 45, pp. 1333–1338, 2015, DOI: 10.3303/CET1545223
- [31] V. Varga *et al.*, “PNS solutions: a P-graph based programming framework for process network synthesis.” *Chemical Engineering Transactions*, vol. 21, pp. 1387-1392, 2010, DOI: 10.3303/CET1021232
- [32] M. Minoux, “Networks synthesis and optimum network design problems: Models, solution methods and applications,” *Networks*, vol. 19, n° 3, pp. 313–360, 1989. DOI: 10.1002/net.3230190305
- [33] L.R. Ford and D.R. Fulkerson, *Flows in Networks*. Princeton, NJ: Princeton University Press, 1962, pp. 1-210.
- [34] A.M. Turing, “Rounding-off Errors in Matrix Processes,” *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 1, n° 1, pp. 287–308, 1948. DOI: 10.1093/qjmam/1.1.287
- [35] R.R., Howell, *Algorithms: A Top-Down Approach*, 9a ed., Dept. of Computing and Information Sciences: KS, Kansas State University, 2009, pp. 1-617.
- [36] D.E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, 3a ed., Redwood City: CA, Addison Wesley Longman Publishing Co., 1997, 672 pp.
- [37] M.T. Goodrich and R. Tamassia, *Algorithm Design and Applications*, 1st ed., Hoboken: NJ, Wiley Publishing, 2014, pp. 1-816.
- [38] A.V. Aho *et al.*, *The Design and Analysis of Computer Algorithms*. Reading: MA, Addison Wesley, 1974, pp. 1-470.
- [39] M.J. Barany (2015). *Industrial Application of the P-graph Framework*, PhD. Dissertation, University of Pannonia, Veszprem, Hungary.
- [40] B. Imreh, J. Fülöp, and F. Friedler, “A note on the equivalence of the set covering and process network synthesis problems”, *Acta Cybernetica*, 14(3), 2000, 497-502.
- [41] R. Karp, *Reducibility among combinatorial problems*. In R. Miller & J. Thatcher (ed.), *Complexity of Computer Computations*, Plenum Press, 1972, pp. 85-103.
- [42] L. Han *et al.*, “FireGrid: an e-infrastructure for next-generation emergency response support”, *Journal of Parallel and Distributed Computing* 70(11) (2010) 1128–1141.

- [43] K. Katayama *et al.*, “Evacuation guidance support using cooperative agent-based IoT devices,” *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, Nagoya, 2017, pp. 1-2, DOI: 10.1109/GCCE.2017.8229431
- [44] I.A. Zualkernan, *et al.*, “An IoT-based Emergency Evacuation System,” *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, BALI, Indonesia, 2019, pp. 62-66, DOI: 10.1109/IoTaIS47347.2019.8980381
- [45] K. Kozminski and E. Kinnen, “An Algorithm for Finding a Rectangular Dual of a Planar Graph for Use in Area Planning for VLSI Integrated Circuits,” *21st Design Automation Conference Proceedings*, Albuquerque, NM, USA, 1984, pp. 655-656.
- [46] A.Y. Lvov *et al.*, “Integer Arithmetic Method for Wire Length Minimization in Global Placement with Convolution Based Density Penalty Computation,” U. S. Patent 10 528 695, Jan 30, 2020.
- [47] S., Karimullah *et al.*, *Floorplanning for Placement of Modules in VLSI Physical Design Using Harmony Search Technique*. In A. Kumar, M. Paprzycki, & V. Gunjan (ed.), ICDSMLA 2019. Lecture Notes in Electrical Engineering, vol 601, Springer, Singapore, 2020.
- [48] T.-C. Chen and Y.-W. Chang, “Modern floorplanning based on B/sup */-tree and fast simulated annealing,” in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 4, pp. 637-650, April 2006, DOI: 10.1109/TCAD.2006.870076