Análisis del Problema de Asignación Generalizada (GAP) con algoritmos genéticos

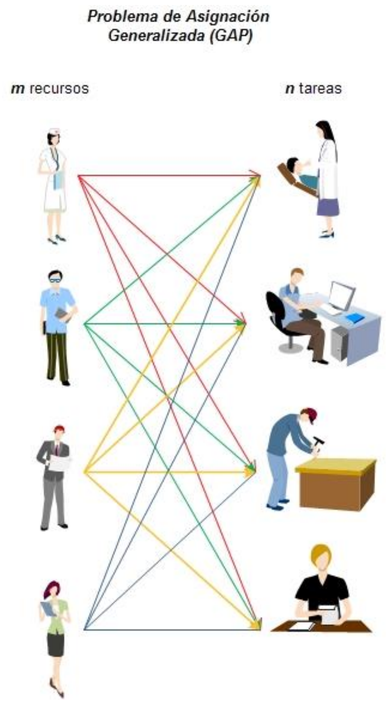
Durán Sevilla Miguel_Ángel, Hernández Santillán Ilsen_Magaly y Minor Popocatl Hertwin.
Departamento de Ingeniería Industrial.
Universidad Politécnica de Tulancingo. Calle Ingenierías No. 100, Huapalcalco, Hidalgo. México
E-mail: mike_durant_s7@hotmail.com, ilsen_magaly@hotmail.com y hertwin.minor@upt.edu.mx

RESUMEN

En este trabajo se revisan aspectos acerca de las decisiones sobre la asignación de recursos en el ámbito de la gestión sanitaria, estos son problemas altamente complejos y que requieren modelos y métodos sofisticados para su solución e implementación. En particular, el Problema de Asignación Generalizada (GAP) es un modelo cuantitativo muy utilizado para gestionar la asignación de recursos en el campo sanitario, sobre todo en las asignaciones correctivas y de determinados equipos o materiales a tareas. El GAP es un problema clasificado como NP-difícil en el ámbito de optimización combinatoria. El objetivo de este trabajo es el estudio para la resolución de este tipo de problema utilizando una heurística que ha probado ser muy eficiente en optimización combinatoria, los Algoritmos Genéticos. Este enfoque es implementado en Matlab y se prueba resolviendo diversas instancias.

Introducción

El Problema de Asignación Generalizada (GAP) [1] es un modelo cuantitativo que se utiliza para gestionar la asignación de *n* tareas a *m* recursos y/o máquinas, por ejemplo, una adecuada asignación de tareas puede representar un ahorro significativo en los gastos del sistema sanitario en la asignación de intervenciones quirúrgicas a quirófanos, aparatos sofisticados de diagnóstico, entre otros; para llevar a cabo dichas actividades se necesita un determinado recurso que puede ser personal médico, laboratorios o en general equipos y materiales. Dichos recursos están disponibles en una cantidad limitada, cada tarea debe ser asignada sólo a un recurso y en muchos casos tienen una elevada utilización y un alto costo. Este tipo de problema se considera NP-Hard [2].



Objetivo

El objetivo de este trabajo es el estudio de la implementación de la heurística Algoritmos Genéticos para el Problema de Asignación Generalizada (GAP)

Modelo matemático

Índices :

I: conjunto de recursos $(i = 1,...,m)$

J: serie de tareas $(j = 1,...,n)$

Parámetros :

$a_i =$ *capacidad del recurso i*

$b_{ij} =$ *consumo si la tarea j es asignada al recurso i*

$c_{ij} =$ *costo de la tarea j si es asignada al recurso i*

Variables :

La variable de decisión es

$$x_{ij} \begin{cases} 1 \text{ si la tarea } j \text{ es asignada al recurso } i, \\ 0 \text{ en otro caso.} \end{cases}$$

$$Min \quad z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s.a

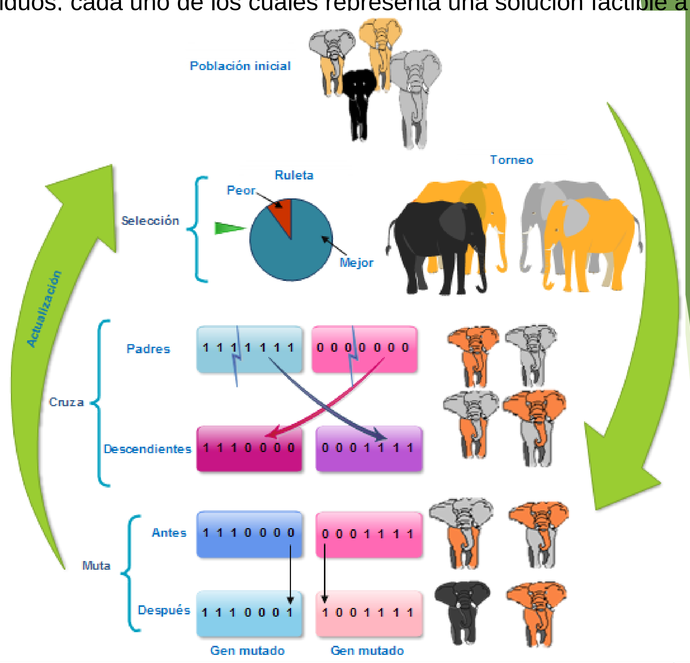
$$\sum_{j=1}^{n} b_{ij} x_{ij} \le a_i, \quad \forall \ i \in M$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad \forall \ j \in N$$

$$x_{ij} \in \{0,1\}, \quad \forall \ i \in M; \forall \ j \in N$$

Heurística

Los Algoritmos Genéticos usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos. cada uno de los cuales representa una solución factible a un problema dado



Algoritmo genético

Inicio (1)
  t = 0
  inicializar P(t)
  evaluar P(t)
  Mientras (no se cumpla la condición de parada) hacer
  Inicio(2)
    t = t + 1
    seleccionar P'(t) desde P(t-1)
    P''(t) ←cruce P'(t)
    P'''(t) ←mutación P''(t)
    P(t) ←reemplazamiento (P(t-1),P'''(t))
    evaluar P(t)
  Final(2)
Final(1)

Experimento computacional

Los parámetros que se manipularon fueron los siguientes:

- Tipo de generación de la población.
- Población inicial
- Tipo de selección.
- Probabilidad de selección.
- Tipo de cruza.
- Probabilidad de cruza.
- Tipo de mutación.
- Probabilidad de mutación.
- Condiciones de parada

Resultados

La instancia que se utilizó para realizar diferentes combinaciones fué de tamaño *m* = 11 y *n* = 11; para dicho problema se consideranon los siguientes parámetros como fijos:

- Generación de la población aleatoria.
- Tamaño de la población = 100.
- Mutación Gaussiana (Uniforme).
- Criterios de paro:
  - 120 generaciones.
  - Limite de generaciones = 10.

	Combinaciones	Selección		Cruza		Mutación	F.O promedio	T promedio (segundos)
		Tipo	Probabilidad	Tipo	Parámetro	Probabilidad		
Mejor	1	TORNEO	0.75	EN UN PUNTO	5	0.05	699.5671125	5.333284375
	2	TORNEO	0.9	EN UN PUNTO	5	0.06	699.687725	5.481485125
	3	TORNEO	0.75	EN UN PUNTO	5	0.09	701.600775	4.54547925
Medio	29	RULETA	0.9	EN UN PUNTO	5	0.07	710.273075	4.74438025
	30	TORNEO	0.9	EN UN PUNTO	5	0.09	710.6807625	4.512329
	31	TORNEO	0.7	EN UN PUNTO	5	0.09	711.2202625	4.742430375
Peor	58	RULETA	0.75	EN UN PUNTO	5	0.09	717.2595875	4.859431125
	59	RULETA	0.7	EN UN PUNTO	5	0.07	718.9271875	4.594229375
	60	RULETA	0.75	EN UN PUNTO	5	0.08	719.680325	4.564979375

CONCLUSIONES

Cuando se incrementa la probabilidad en la mutación se obtienen peores soluciones, en otras palabras, esa pequeña alteración no produce características deseables y esto se debe a que se introduce una variedad genética excesiva. También se obtuvo que al utilizar la cruza en un punto se obtuvieron mejores soluciones en comparación que al utilizar cruza en dos puntos (estos diferentes tipos de cruza se realizaron a priori).

Dados los resultados obtenidos se tuvo que la mejor combinación de todas las que se realizaron en este experimento fue la combinación número 1, la cual se mostró anteriormente en la tabla de resultados.

Entonces como conclusión general se puede decir que la Heurística AG es un algoritmo eficiente para resolver problemas de tipo NP-Hard, ya que al implementar adecuadamente este algoritmo se obtienen buenas soluciones y el tiempo computacional de procesamiento es relativamente pequeño, y esta última parte es muy importante para las organizaciones.

Para trabajos futuros se consideran resolver otros tipos de problema del tipo NP-Hard implementando esta Heurística.

REFERENCIAS

[1] Chu, P.C. and J.E. Beasley (1997). «A Genetic Algorithm for the Generalized Assignment Problem.» Computers and Operations Research.

[2] L. Alfandari, A. Plateau and P.Tolla, (2004) A path relinking algorithm for the generalized assignment problem, in M.G.C. Resende, J.D.Sousa (Eds.), Metaheuristics: Computer Decision-Making, Kluwer Academic Publishers, Beston.

[3] M. Yagiura, T. Ibaraki and F.Glover (2004). An ejection chain approach for the generalized assignment problem, Informs Journal of Computing 16.

[4] S. Martello and P. Toth (1990) Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York.