

Estrategias evolutivas para la minimización del *makespan* en una máquina con tiempos de preparación dependientes de la secuencia

Evolution Strategies to Minimize the Makespan in a Single Machine with Sequence Dependent Setup Times

Salazar-Hornig Eduardo

Facultad de Ingeniería
Universidad de Concepción, Concepción, Chile
Correo: esalazar@udec.cl

Schrils-Abreu Giselle

Facultad de Ingeniería
Universidad de Concepción, Concepción, Chile
Correo: gischrils@udec.cl

Información del artículo: recibido: enero de 2011, aceptado: marzo de 2013

Resumen

En este trabajo se presenta una aplicación de un algoritmo de estrategia evolutiva multi-miembro ($\mu+\lambda$) – ES para la programación de trabajos en una máquina con tiempos de preparación dependientes de la secuencia con el objetivo de minimizar el *makespan* (C_{max}). La estrategia evolutiva fue evaluada sobre un conjunto de problemas generados en forma aleatoria. Se introduce un procedimiento de mejora de la estrategia evolutiva, generando la población inicial como una vecindad de la solución entregada por otro método, lo que mejora su desempeño. La estrategia evolutiva se comparó con la heurística del mejor vecino y un algoritmo genético, mostrando un mejor desempeño.

Abstract

A multi-member ($\mu+\lambda$) – ES evolution strategy algorithm for the single machine scheduling problem with sequence dependent setup times and makespan (C_{max}) minimization is presented. The evolution strategy is evaluated on a random generated set of test problems. A procedure to improve the performance of the evolution strategy considering the initial population as a neighborhood of the solution given by another method is introduced. The evolution strategy shows better performance than a greedy constructive heuristic and a genetic algorithm.

Descriptores:

- una máquina
- estrategias evolutivas
- *makespan*
- algoritmos genéticos
- mejor vecino

Keywords:

- single machine
- evolution strategies
- makespan
- genetic algorithm
- best neighbor

Introducción

La programación de órdenes de producción para la generación de bienes y servicios ayuda a la optimización de recursos productivos limitados satisfaciendo en forma eficiente los requerimientos de los clientes. De esta forma la programación de la producción se ha convertido en una herramienta estratégica para cualquier organización, transformando las necesidades de los clientes en órdenes de producción que se “transforman” en trabajos con fecha de entrega asociada.

Los sistemas de producción por lotes se caracterizan por la obtención de múltiples productos en lotes de producción utilizando la misma instalación, las que se clasifican según la variedad y homogeneidad de los productos a fabricar, y el número de máquinas del sistema (Baker, 1974; Pinedo, 2008). El problema de programación de un *taller de una máquina* consiste en un sistema de una máquina, que procesa n trabajos de una operación procesados en secuencia, eventualmente, con tiempos de preparación (*setup*) dependientes de la secuencia.

Se han aplicado varias heurísticas y algoritmos a diferentes problemas de programación de una máquina con *setup*. Lee y Asllani (2004) comparan la programación matemática entera con algoritmos genéticos para minimizar el número de trabajos atrasados y el *makespan*. Gupta y Smith (2006) proponen algoritmos GRASP y de búsqueda en una vecindad para minimizar la tardanza, mientras que Liao y Juan (2007) proponen un algoritmo ACO para resolver el mismo problema y comparan con otros algoritmos. Koulamas y Kyparisis (2008) tratan el problema con *setups* proporcionales al tiempo de proceso optimizando objetivos relacionados con el tiempo de finalización de los trabajos, entre ellos el *makespan*, aplicando algoritmos de ordenamiento. Lai y Lee (2010) analizan el problema de minimizar el *makespan* y la tardanza aplicando una función no lineal de deterioro del tiempo de proceso, mientras que Salazar y Sánchez (2011) aplican MMAS (una versión de algoritmo ACO) para minimizar el *makespan*. Jula y Rafiey (2012) consideran ventanas de tiempo para el inicio del proceso de cada trabajo, cuidando de mantener cierto nivel de trabajo en proceso (objetivo primario) y de minimizar el *makespan* (objetivo secundario).

Las estrategias evolutivas (ES) fueron introducidas por Rechenberg y Schwefel a mediados de la década del 60 orientadas a la optimización continua de parámetros en problemas de ingeniería (Rechenberg, 1973). En la literatura se encuentran pocas propuestas que adaptan estrategias evolutivas a problemas de secuenciación de trabajos. Filipic y Zupanic (1999) utilizan es-

trategias evolutivas para programar cortes de energía en un sistema productivo con el objetivo de minimizar el consumo de energía. Hou y Chang (2002) proponen un método de estrategias evolutivas para asignar la producción en un sistema productivo multiplanta. Pierreval *et al.* (2003) y Salazar y Rojas (2010) tratan problemas de diseño y configuración de sistemas de producción utilizando estrategias evolutivas. Probablemente los trabajos de Ablay (1987), Herdy (1991) y Rudolph (1991) son los primeros trabajos (y tal vez los únicos) en proponer estrategias evolutivas para resolver el problema del vendedor viajero.

Problema de programación de una máquina

En este trabajo se trata el problema de programar n trabajos en un *taller de una máquina con setups dependientes de la secuencia*, que consiste en secuenciar los n trabajos de manera que se minimice el *makespan* (C_{\max}), es decir, minimiza el tiempo transcurrido entre el inicio del procesamiento del primer trabajo (tiempo de referencia 0) y el tiempo de finalización del procesamiento del último trabajo. El tiempo de proceso de cada trabajo es fijo y existen tiempos de preparación de máquina que dependen del orden en el que se procesan los trabajos en esa máquina. Se consideran los siguientes supuestos:

- El tiempo de proceso del trabajo i está dado por p_i ($i = 1, \dots, n$).
- El tiempo de preparación (*setup*) para procesar el trabajo j después de procesar el trabajo i está dado por s_{ij} ($i = 1, \dots, n; j = 1, \dots, n$), donde s_{ii} representa la preparación inicial cuando el trabajo i es el primer trabajo procesado en la máquina.
- El proceso de un trabajo en la máquina no se puede interrumpir (*non preemption*).
- Todos los trabajos son independientes entre sí y se encuentran disponibles en el instante inicial.
- La máquina opera sin fallas en el horizonte de programación.
- El objetivo es minimizar C_{\max} .

Con la notación introducida por Graham *et al.* (1979), el problema de una máquina caracterizado por los supuestos mencionados se denota por $1|s_{ij}|C_{\max}$ y es un conocido problema NP-Hard (Blazewicz *et al.*, 1996 y Pinedo, 2008), lo que hace impracticable la obtención de la solución óptima para problemas de mediano a gran tamaño (este problema tiene una estructura similar al problema clásico del agente viajero asimétrico ATSP que es una variante del problema del agente viajero en la que las distancias son asimétricas, es decir, la distan-

Download English Version:

<https://daneshyari.com/en/article/274969>

Download Persian Version:

<https://daneshyari.com/article/274969>

[Daneshyari.com](https://daneshyari.com)